

Model Curriculum for Three/Four Year Degree Course
(With Multiple Entry /Exit Option)
Based on NEP-2020

INFORMATION TECHNOLOGY



Odisha State Higher Education Council, Bhubaneswar
Government of Odisha

SEMESTER-I

PAPER I: PROBLEM SOLVING USING C PROGRAM

UNIT- I

Algorithm, Flowchart, Logic Development & Problem Solving. Pseudocode, Control structures (sequence, selection, Iteration), Testing and debugging. Introduction to Programming Language: Introduction to C Programming, Character Set, C Tokens, Keywords & Identifiers, Constants, Variables, Data Types, Variables, Storage Classes. Operators and Expressions: Properties of operators, Priority of operators, Comma and conditional operator, Arithmetic operators ,Relational operators, Assignment operators and expressions, Logical Operators, Increment & Decrement operators ,Conditiona l& Bitwise operators ,Expressions ,Input and Output Operations in C.

UNIT- II

Decision Making and Branching: Simple IF Statement, IF ELSE Statement, Nesting IF ELSE Statement, ELSE IF Ladder, Switch Statement, Operator, GOTO Statement. Decision Making and Looping: The WHILE Statement, The DO Statement, The FOR Statement, Jumps in LOOPS. Arrays: Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two- dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Passing objects in Array.

UNIT- III

User-defined Functions: Need, Elements & Definition, Function Calls, Function Definition, Category of Functions, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments, Recursion. Pointers: Accessing the Address of a Variable, Declaring Pointer Variables, Initializations of Pointer Variable, Accessing a Variable through its Pointer, Chain of Pointers, Pointer Expressions, Pointer Increments and Scale Factor, Pointers and Arrays, Pointers and Character Strings, Array of Pointers, Pointers as Function Arguments, Functions Returning Pointers, Pointers to Functions, Pointers to Structures, Troubles with Pointers, Pointers variables.

UNIT- IV

Structures and Unions: Defining, Declaring, Accessing, Initialization Structure, Arrays of Structures, Arrays within Structures, Structures and Functions, Unions, Passing structure to function, Pointers to structures. File Management in C: Defining and Opening a File, Closing a File, Input/ Output Operations on Files, standard I/O, Formatted I/O, Error Handling during I/O Operations, Random Access to Files, Command Line Arguments. Memory Allocation: Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation.

TEXT BOOK:

1. E. Balaguruswamy “Programming in ANSIC”, Tata McGraw Hill.

REFERENCE BOOKS:

1. B. Kernighan & D.M.Ritche, “The C Programming Language”, PHI
2. Byron Gottfried, “Programming with C”, TMH Publications.

LAB: PROBLEM SOLVING USING C PROGRAM

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series $S = 1 + 1/2 + 1/3 + 1/4 + \dots$.
4. WAP to compute the sum of the first n terms of the following series $S = 1 - 2 + 3 - 4 + 5 - \dots$.
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers .WAP to use it.
9. WAP to find out greatest number without using third variable.
10. WAP to find out Permutation and Combination of numbers.
11. WAP to print a triangle of stars as follows(take number of lines from user):

```
      *
     ***
    *****
   ********
  **********
 **********
```

12. WAP to perform following actions on an array entered by the user:
 - i) Print the even-valued elements
 - ii) Print the odd-valued elements
 - iii) Calculate and print the sum and average of the elements of array
 - iv) Print the maximum and minimum element of array
 - v) Remove the duplicates from the array
 - vi) Print the array in reverse order The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
13. WAP that print a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
14. Write a program that swaps two numbers using pointers.
15. Write a program in which a function two variables and then alter its contents.
16. Write a program which takes the radius of a circle a sin put from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
17. Write a program to find sum of n elements entered by the user. To write this program, Allocate memory dynamically using malloc () / calloc () functions or new operator.
18. Write a program to perform following operations on strings:
 - a) Show address of each character in string
 - b) Concatenate two strings without using struct function.
 - c) Concatenate two strings using struct function.
 - d) Compare two strings.
 - e) Calculate length of the string(use pointers).
 - f) Calculate length of the string(use pointers).
 - g) Convert all lower case characters to uppercase.
 - h) Convert all upper case characters to lowercase.
 - i) Calculate number of vowels.
 - j) Reverse the string.
19. Given two ordered array of integers, write a program to merge the two-arrays to get an ordered array.

20. WAP to display Fibonacci series (i) using recursion, (ii) using iteration.
21. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration.
22. WAP to calculate GCD of two numbers (i) with recursion, (ii) without recursion.
23. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum, b) Difference, c) Product, d) Transpose.
24. Copy the contents of one text file to another file, after removing all whitespaces.

PAPER II: COMPUTER ORGANIZATION AND ARCHITECTURE

UNIT-I

Introduction to Computers: Hardware, Software, Types of Software, Introduction to OS. GUI vs CUI. Data Representation and Basic Computer Arithmetic :Number systems, Decimal, Binary, Octal and Hexadecimal number systems and their inter conversion, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison,multiplication and division algorithms for integers,Overflow and Underflow,Character Codes, Boolean Algebra, Electronic Logic Gates, Synthesis of Logic Functions, Minimization of Logic Expressions, Minimization using Karnaugh Maps, Synthesis with NAND and NOR Gates.

UNIT-II

Classification to Programming Language :Machine language, Assembly language, 3rd&4th generation languages, Characteristic of 3 GI 's and 4 GL's, Assemblers and Compilers. Basic Logic Functions: Flip-Flops, Gated Latches, Master-Slave Flip-Flops, Edge-Triggering, T Flip- Flops, JK Flip-Flops. Registers and Shift Registers, Counters, Decoders, Multiplexers, Programmable Logic Devices (PLDs), Programmable Array Logic (PAL), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Array (FPGA), Sequential Circuits, Timing Diagrams, The Finite State Machine Model, Synthesis of Finite State Machines.

UNIT-III

Basic Structure of Computers: Computer Types, Functional Units, Input Unit, Memory Unit, Arithmetic and Logic Unit, Output Unit, ControlUnit , Basic Operational Concepts, Bus Structures, Software. Memory System: Semiconductor RAM Memories, Internal Organization of Memory Chips, Static Memories, Asynchronous DRAMS, Synchronous DRAMS, Structure of Large Memories, Memory System Considerations, RAMBUS Memory. Read-Only Memories: ROM, PROM, EPROM, EEPROM, Flash Memory, Speed, Size, and Cost of Memory. Secondary Storage: Magnetic Hard Disks, Optical Disks, Magnetic Tape Systems.

UNIT-IV

Machine Instructions and Programs : Numbers, Arithmetic Operations, and Characters: Number Representation, Addition of Positive Numbers, Addition and Subtraction of Signed Numbers, Overflow of Integer Arithmetic, Characters, Memory Locations and Addresses, Byte Addressability, Word Alignment, Accessing Numbers, Characters, and Character Strings, Memory Operations, Instructions and Instruction Sequencing, Register Transfer Notation, Basic Instruction Types, Instruction Execution and Straight-Line Sequencing, Branching, Condition Codes, Generating Memory Addresses, Addressing Modes, Implementation of Variables and Constants, Indirection and Pointers, Indexing and Arrays, Relative Addressing.

TEXT BOOKS:

1. William Stallings, "Computer Organization and Architecture", Pearson Education
2. Morris M. Mano, "Computer System Architecture", PHI New Delhi.

REFERENCE BOOKS:

1. Carl Hamacher, Z. Vranesic, S. Zaky, "Computer Organization", 5/e (TMH)
2. B. Ram, "Computer Fundamentals", New Age International Publishers.

LAB: COMPUTER ORGANIZATION AND ARCHITECTURE

List of Practicals using VHDL/Verilog Simulation

1. Simulation of Logic gates (AND, OR, NOT, XOR) of two 8-bit numbers.
2. Simulation of Universal gates of two 8-bit numbers.
3. Add two 8-bit numbers (with carry bit, without carry bit).
4. Subtract two 8-bit numbers.
5. Add two 16-bit numbers (with carry bit, without carry bit).
6. Subtract two 16-bit numbers.
7. Design and implement half-adder circuit.
8. Design and implement full-adder circuit.
9. Design and implement a 2:1 multiplexer.
10. Design and implement a 4:1 multiplexer.
11. Design and implement a S-R, D, T-flip flop.
12. Design and implement a 3-bit counter.

SEMESTER-II

PAPER III: DATA STRUCTURE

UNIT- I

Basic definitions: Classifications, ADT; Linear Data Structures - Sequential representations, Non-linear data structures – representations Arrays: Single and Multi-dimensional Arrays; Sparse Matrices (Array and Linked Representation); Row major and column-major order; different applications. Linked Lists: Singly, Doubly and Circular Lists (Array and Linked representation), Normal and Circular representation of Stack in Lists, Self-Organizing Lists; Skip Lists

UNIT-II

Stacks: Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Evaluation of postfix expression using stack; Applications of stack; Limitations of Array representation of stack Queues: Array and Linked representation of Queue, De-queue, Priority Queues Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation); Tail Recursion; When not to use recursion; Removal of recursion.

UNIT-III

Trees and Graphs: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion , Recursive and Iterative Traversals on Binary Search Trees); Different Properties of Binary trees; Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees); B- tree, B+ tree; Graphs - Representations, Breadth-first and Depth- first Search. Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search; Bubble Sort, Selection Sort, Insertion Sort, Heap Sort, Merge Sort, Radix Sort, Bucket Sort, Shell Sort; Comparison of Sorting Techniques.

UNIT-IV

Hashing: Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table, Reordering, Resolving collision by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing Function. File Structures: Sequential and Direct Access. Relative Files, Indexed Files -B+tree as index. Multi-indexed Files, Inverted Files, Hashed Files.

TEXT BOOKS:

1. D. Samanta, “Classic data Structures”, PHI.

REFERENCE BOOKS:

1. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, “Data Structures Using C and C++”, Second Edition, PHI, 2009.
2. Robert L. Kruse, "Data Structures and Program Design in C", Pearson.

LAB: DATA STRUCTURE

LIST OF PROGRAMMES (USING C /C ++)

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
 5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
 9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomials.
11. WAP to calculate factorial and to compute the factors of a given no. (i) using recursion, (ii) using iteration
12. WAP to display Fibonacci series (i) using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 numbers (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree:
 - (a) Insertion (Recursive and Iterative Implementation)
 - (b) Deletion by copying
 - (c) Deletion by Merging
 - (d) Search a no. in BST
 - (e) Display its preorder, postorder and inorder traversals Recursively
 - (f) Display its preorder, postorder and inorder traversals Iteratively
 - (g) Display its level-by-level traversals
 - (h) Count the non-leaf nodes and leaf nodes
 - (i) Display height of tree
 - (j) Create a mirror image of tree
 - (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per in order traversal, and implement operations like finding the successor / predecessor of an element, insert an element, in order traversal.
23. WAP to implement various operations on AVL Tree.

PAPER IV: PYTHON PROGRAMMING

UNIT-I:

Python: Features of Python , Installing Python for windows and setting up paths, writing and Executing of a python programs, Python Virtual machine, Frozen binaries, Comparison between C,Java and python, Comments,Doc strings,How python sees variables,Data types in Python,built in types,sequences in python,sets,literals in Python,user defined datatypes, identifiers &reserved words, Naming convention in python.

UNIT-II:

Various Operators in Python, Input& Output, Control statements, if statements, whileloop ,for loop, infinite loop, nested loop, else suit, break, continue, pass, assert, return statements, command line arguments.

Arrays in python, advantages using arrays, creating arrays, importing the array module,indexing and slicing on arrays,Processing the arrays,Comparing arrays.

Strings in Python, creating strings, Length of a string, indexing in strings, slicing strings,Concatenation and Comparing strings,Finding Sub Strings, Replacing a String.

UNIT-III:

Functions in Python,define a function,calling a function,return from function,pass by object Reference,Positional arguments, Default arguments Recursive

functions. Introduction to OOP, features of OOP, creating classes,thelself-variable,constructor,types of variables,name spaces,types of methods.

UNIT-IV:

Inheritance:Define inheritance,types of inheritance,and constructors ininheritance,overriding superclass constructors & methods, thesuper () method ,MRO Polymorphism: Duck typing philosophy of Python, operatoroverloading, methodoverriding, interfacesin python.

Exceptions: Errors in a python program, Exceptions, Exception handling, Types of Exceptions,the Exception block,the assert statement, user defined exceptions.

TEXT BOOKS:

- T. Budd, Exploring Python,TMH,1stEd, 2011.
- Core Python Programming, Dr. R. Nageswar Rao, Dreamtech Press.
- Python Programming for Absolute Beginners, Michael Dawson, Cengage Learning.

REFERENCE BOOKS:

- ✓ Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist: Learning with Python,Freelyavailable online.2012.

LAB: PYTHON PROGRAMMING

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
2. Write a Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria: Grade A:Percentage \geq 80, Grade B:Percentage \geq 70 and $<$ 80Grade C:Percentage \geq 60 and $<$ 70Grade D:Percentage \geq 40 and $<$ 60 GradeE:Percentage $<$ 40.
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. Write a Program to display the first n terms of Fibonacci series.
5. Write a Program to find factorial of the given number.
6. Write a Program to find sum of the following series for n terms: $1 - 2/2! + 3/3! \dots$
7. Write a Program to calculate the sum and product of two compatible matrices.

SEMESTER-III

PAPER V: DATABASE MANAGEMENT SYSTEM

UNIT- I

Fundamental of DBMS:Data,Information,Database & Computers.DBMS definition,DBMS versusfile processing system, Components of DBMS Environment, Instances &Schemas, Three Levels Architecture, Data Independence, Data Dictionary, Database Users, Data Administrators, Data Modelling Using the Entity-Relationship(ER) Model.

UNIT- II

Modelling the Real World, Various Data Models & their comparison, EntityRelationship Models, RDBMS – Concept,Components,Data Integrity,Constraints,Keys,Relational Data Manipulations, Relational Algebra and Calculus.

UNIT- III

Relational Database Design by ER and EER to Relational Mapping, Introduction to SQL,DDL,DML and DCL statements, Creating tables, Adding Constraints, Altering Tables, Update, Insert, Delete & Various Forms of SELECT – simple, Using Special Operators for Data Access, Nested Queries & Exposure to Joins, Aggregate Functions.

UNIT- IV

Functional Dependencies and Normalization for Relational Database Normalization: Definition, Decomposition, Basic concepts like FD,Objectives of Normalization,NormalForms–First, Second and Third Normal Form, BCNF, Concept of Multi-valued dependencies.

Transaction: Concurrency Control - Problem & its Basis, Concurrency Control-Locks & Deadlocks, Recovery – Kind of Failures, Recovery Techniques, Security- Authentication, Authorization, Access Control.

TEST BOOKS:

1. R. Elmasri, S. B. Navathe,“Fundamentals of Database Systems”, 6/e,PearsonEducation.
2. A. Silberschatz, H. F. Korth, S. Sudarshan,“Database System Concepts”, 6/e, McGrawHill.

REFERENCE BOOKS:

1. C. J. Date,An Introduction to Database Systems, Pearson Education, NewDelhi.
2. C. Coronel, S. Morris, & P. Rob, “Database Principles (Fundamentals of Design, Implementation, and Management)”, 9/e, Cengage Learning.
3. I van Bayross, SQL, PL/SQL, BPB Publication, NewDelhi

LAB: DATABASE MANAGEMENT SYSTEM

Create and use the following data base schema to answer the given queries.

EMPLOYEE Schema

Field	Type	NULL	KEY	DEFAULT
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL
Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commission	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

DEPARTMENT Schema

Field	Type	NULL	KEY	DEFAULT
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL
Location	Varchar(50)	Yes		New Delhi

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than \$2850. .
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of \$1500 and \$2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is 'A'.
14. Query to display Name of all employees either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Mangers Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.

16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1stMonday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants <3*Current Salary>. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Department Name of all employees who have an 'A' in their name.
25. Query to display Name, Job, Department No. and Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees Name who do not have a Manager.
27. Query to display Name, Department No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies \$100
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees.
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more thanthe average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.
36. Query to display the names and salaries of all employees who report to King. Try the suggestions below or type a new query above.

PAPER VI: DISCRETE MATHEMATICAL STRUCTURE

UNIT-I:

Logic and Proofs: Propositional logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs.

UNIT-II:

Sets: Venn Diagrams, Subsets, The size of a set, Power Sets, Cartesian Products, Set Operations
Functions: One-to-One and Onto Functions, Inverse Functions and Compositions of Functions Partial Functions. Sequences, Recurrence Relations, Summations.

UNIT-III:

Counting: The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients and Identities.

Relations: Relations and their Properties, n-ary Relations and their Applications, Representing Relations, Closure of Relations, Equivalence Relations, partial Orderings.

UNIT-IV:

Graph Terminology and Special Types of Graphs, Bipartite Graphs, Representing Graphs: Isomorphism of Graphs, Euler and Hamilton Paths, Shortest Path Problems: Dijkstra's Algorithm, Traveling Salesperson Problem, Planar Graphs, Graph Coloring. Tree Traversal, Minimum Spanning Trees

Text Books:

- ✓ *Kenneth H. Rosen, Discrete Mathematics and its Applications, Mc Graw Hills International Seventh Edition.*

Reference Books:

- ✓ *Elements of Discrete Mathematics by C. L. Liu and D.P. Mohapatra, TMH, 2012*
- ✓ *J. P Tremblay, R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", TMH, 1997.*

PAPER VII: OPERATING SYSTEMS

UNIT-I:

Introduction to Operating System, Computer System Architecture, System Structures: Operating system services, User and Operating-System Interface, system calls, system programs, Operating system design and implementation, Operating system structure, Batch processing, multi-programming, time-sharing and real-time systems

UNIT-II:

Process Management: Process Concept, Operations on processes, Process scheduling, Inter-process Communication, Threads, Multithreading Models. CPU Scheduling algorithms: Scheduling Criteria, FCFS, SJF, Priority, Round Robin, Multilevel Queue, Multilevel Feedback Queue. Deadlocks: Deadlock detection, deadlock prevention, and deadlock avoidance fundamentals.

UNIT-III:

Memory Management Strategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory Management: Concepts, Demand Paging, Page Replacement techniques: FIFO, LRU, Optimal, Thrashing.

UNIT-IV:

Storage Management: Overview of Mass-Storage Structure, Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK, RAID technology.

File System concept, Access Methods, Directory and Disk Structure, File System systems, File, Sharing and File Protection.

Text Books:

- ✓ Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, Eighth Edition, Wiley Student Edition 2009
- ✓ Operating Systems, Rajiv Chopra, S. Chand Pubs.

Reference Books:

- ✓ Modern Operating System, Tanenbaum, Pearson, 4/ed. 2014
- ✓ Operating Systems 5th Edition, William Stallings, Pearson Education India
- ✓ Richard Blum, Linux Command Line and Shell Scripting Bible, O' Reilly

LAB: OPERATING SYSTEMS

1. Basic Linux Commands and Overview (date, cal, who, tty, echo, bc, pwd, mkdir, rmdir, cd, cat, cp, mv, rm, ls, wc)
2. Shell Programming
 - i. Write a shell script to perform the tasks of basic calculator.
 - ii. Write a shell script to find the greatest number among the three numbers.
3. Shell Programming
 - i. Write a shell script to check if the number entered at the command line is prime or not.
 - ii. Write a shell script to display the multiplication table of any number.
4. Shell Programming
 - i. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
 - ii. Write a shell script to find the sum of digits of a given number.
 - iii. Write a shell script to find the factorial of a given number.
5. Write a program (using fork() and/or exec() commands) where parent and child execute:
 - i. Same program, same code.
 - ii. Same program, different code.
 - iii. Before terminating, the parent waits for the child to finish its task.
6. Write a program to copy files using system calls.
7. Write a program using C to implement FCFS scheduling algorithm.
8. Write a program using C to implement Round Robin scheduling algorithm.
9. Write a program using C to implement SJF scheduling algorithm.
10. Write a program using C to implement first-fit, best-fit, and worst-fit allocation strategies.

SEMESTER-IV

PAPER VIII: JAVA PROGRAMMING

UNIT-I:

Introduction to Java: Java History, Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords (super, this, final, abstract, static, extends, implements, interface) , Data Types, Wrapper class, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods). Input through keyboard using Command line Argument, the Scanner class, BufferedReader class.

UNIT-II:

Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Class Variables & Methods, Objects, Object reference, Objects as parameters, final classes, Garbage Collection. Constructor- types of constructors, this keyword, super keyword. Method overloading and Constructor overloading. Aggregation vs Inheritance, Inheritance: extends vs implements, types of Inheritance, Interface, Up-Casting, Down-Casting, Auto-Boxing, Enumerations, Polymorphism, Method Overriding and restrictions. Package: Pre-defined packages and Custom packages.

UNIT-III:

Arrays: Creating & Using Arrays (1D, 2D, 3D and Jagged Array), Array of Object, Referencing Arrays Dynamically. Strings and I/O: Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality,

Passing Strings To & From Methods, StringBuffer Classes and StringBuilder Classes. IO package: Understanding StreamsFile class and its methods, Creating, Reading, Writing using classes: Byte and Character streams, FileOutputStream, FileInputStream, FileWriter, FileReader, InputStreamReader, PrintStream, PrintWriter. Compressing and Uncompressing File.

UNIT-IV:

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi- threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Text Book:

E. Balagurusamy, "Programming with Java", TMH, 4/Ed

Reference Book:

Herbert Schildt, "The Complete Reference to Java", TMH, 10/Ed.

LAB: JAVA PROGRAMMING

1. To find the sum of any number of integers entered as command line arguments.
2. To find the factorial of a given number.
3. To convert a decimal to binary number.
4. To check if a number is prime or not, by taking the number as input from the keyboard.
5. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument.
6. Write a program that show working of different functions of String and StringBuffer classes like `setCharAt()`, `setLength()`, `append()`, `insert()`, `concat()` and `equals()`.
7. Write a program to create a – “distance” class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
8. Modify the – “distance” class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
9. Write a program to show that during function overloading, if no matching argument is found, then Java will apply automatic type conversions (from lower to higher data type).
10. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword.
11. Write a program to show the use of static functions and to pass variable length arguments in a function.
12. Write a program to demonstrate the concept of boxing and unboxing.
13. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
14. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.
15. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
16. Write a program – “DivideByZero” that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
17. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
18. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
19. Write a program to demonstrate priorities among multiple threads.
20. Write a program to demonstrate different mouse handling events like `MouseClicked()`, `MouseEntered()`, `MouseExited()`, `MousePressed()`, `MouseReleased()` & `MouseDragged()`.
21. Write a program to demonstrate different keyboard handling events.

PAPER IX: SOFTWARE ENGINEERING

UNIT-I:

Introduction: Evolution of Software to an Engineering Discipline, Software Development Projects, Exploratory Style of Software Development, Emergence of Software Engineering, Changes in Software Development Practices, Computer Systems Engineering. Software Lifecycle Models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Agile Development Models, Spiral Model.

UNIT-II:

Software Project Management: Software Project Management Complexities, Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO, Halstead's Software Science, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management.

UNIT-III:

Requirement Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specifications, Formal System Specification Axiomatic Specification, Algebraic Specification, Executable Specification and 4GL.

Software Design: Design Process, characterize a Good Software Design, Cohesion and Coupling, Layered Arrangements of Modules, Approaches to Software Design (Function Oriented & Object-Oriented).

UNIT-IV:

Coding and Testing: Coding: Code Review, Software Documentation, Testing, Unit Testing, Black Box and White Box Testing, Debugging, Program Analysis Tools, Integration Testing, System Testing, Software Maintenance.

Text Book:

Fundamental of Software Engineering, Rajib Mall, Fifth Edition, PHI Publication, India.

Reference Books:

- ✓ Software Engineering– Ian Sommerville, 10/Ed, Pearson.
- ✓ Software Engineering Concepts and Practice – Ugrasen Suman, Cengage Learning India Pvt, Ltd.
- ✓ Software Engineering, R Khurana, Vikash Pubs.

LAB: SOFTWARE ENGINEERING

Students have to do at least two software development projects from the list of projects given below. They have to follow the complete software development lifecycle with the following details. UML can be used as a design tool. (Coding is optional).

1. Problem Statement
2. Process Model
3. Requirement Analysis:
 - Creating a Data Flow
 - Data Dictionary, Use Cases
3. Project Management:
 - Computing FP
 - Effort
 - o Schedule, Risk Table, Timeline Chart
4. Design Engineering:
 - Architectural Design
 - Data Design, Component Level design
5. Testing:
 - Basis Path Testing

List of Projects:

1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers.
2. Route Information: Online information about the bus routes and their frequency and fares
3. Car Pooling: To maintain a web-based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.

PAPER X: CLOUD COMPUTING

UNIT-I:

Overview of Computing Paradigm - Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing - Evolution of cloud computing - Cloud Computing (NIST Model) Characteristics - Pros and Cons of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing – Role of Open Standards - Cloud computing stack - Service Models (XaaS).

UNIT-II:

Cloud Computing Architecture: Layers in cloud architecture, Software as a Service (SaaS), features of SaaS and benefits, Platform as a Service (PaaS), features of PaaS and benefits, Infrastructure as a Service (IaaS), features of IaaS and benefits, Service providers, challenges and risks in cloud adoption. Cloud deployment model: Public clouds – Private clouds – Community clouds - Hybrid clouds.

UNIT-III:

Infrastructure as a Service (IaaS) – Introduction- IaaS definition, virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM) - Resource Virtualization – Server, Storage, Network, Virtual Machine(resource) provisioning and manageability, Data storage in cloud computing (storage as a service) - Examples - Amazon EC2 - Renting, EC2 Compute Unit.

Platform as a Service (PaaS) – Introduction, Service Oriented Architecture (SOA) - Cloud Platform and Management – Computation, Storage – Examples - Google App Engine, Microsoft Azure, Salesforce.com, Force.com platform - Software as a Service (SaaS) - Introduction to SaaS - Web services, Case Study on SaaS.

UNIT-IV:

SLA Management in Cloud Computing: Traditional Approaches to SLO Management, Types of SLA, Life Cycle of SLA, SLA Management in Cloud, Pricing Mechanism, SLA Violation.

Cloud Security - Infrastructure Security - Network level security, Host level security, Application- level security – Data security and Storage - Data privacy and security Issues, Jurisdictional issues raised by Data location - Identity & Access Management - Access Control - Trust, Reputation, Risk - Authentication in cloud computing, Client access in cloud, Cloud contracting Model.

Text Books:

- ✓ Cloud computing a practical approach - Anthony T.Velte , Toby J. Velte Robert Elsenpeter TATA McGraw- Hill , New Delhi.
- ✓ Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley.
- ✓ Cloud Computing, U S Pandey & K Choudhary, S. Chand Pubs.

Reference Books:

- ✓ Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall.
- ✓ Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer.

SEMESTER-V

PAPER XI: COMPUTER NETWORK

UNIT-I:

Introduction to Data Communications and Network Models: Protocols and Standards, Layers in OSI Models, Analog and Digital Signals, Transmission Modes, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission, Network Devices & Drivers: Router, Modem, Repeater, Hub, Switch, Bridge (fundamental concepts only).

UNIT-II:

Signal Conversion: Digital-to-Digital Conversion, Analog-to-Digital Conversion, Digital-to-analog Conversion, Analog-to-Analog Conversion. Transmission Media: Guided Media, Unguided Media, Switching Techniques: Packet Switching, Circuit Switching, Datagram Networks, Virtual-Circuit Networks, and Structure of a Switch.

UNIT-III:

Error Detection and Correction: Checksum, CRC, Data Link Control: Framing, Flow and Error Control, Noiseless Channels, Noisy channels, (Stop and Wait ARQ, Sliding Window Protocol, Go Back N, Selective Repeat) HDLC, Point-to-Point Protocol. Access Control: TDM, CSMA/CD, and Channelization (FDMA, TDMA, and CDMA).

UNIT-IV:

Network Layer: Logical Addressing, IPv4 Addresses, IPv6 Addresses, Virtual-Circuit Networks: Frame Relay and ATM, Transport Layer: Process-Process Delivery: UDP, TCP. Application layers: DNS, SMTP, POP, FTP, HTTP, Basics of WiFi (Fundamental concepts only), Network Security: Authentication, Basics of Public Key and Private Key, Digital Signatures and Certificates (Fundamental concepts only).

Text Book:

Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH.

Reference Book:

Computer Networks, A. S. Tanenbaum, 4th edition, Pearson Education.

LAB: COMPUTER NETWORK

1. Use the `ipconfig` (Windows) or `ifconfig` (Linux/Mac) command to display the current network configuration.
 - i. Identify and document the IP address, subnet mask, and default gateway of the system.
 - ii. Change the IP address of the system using `netsh` (Windows) or `ifconfig` (Linux/Mac). Verify the change using the same command.
 - iii. Experiment by configuring static IP, dynamic IP.
2. Use the `ping` command
 - i. to check connectivity between Systems in your Lab.
 - ii. to a remote server (e.g., `google.com`).
 - iii. Analyze the round-trip time and packet loss.
3. Use the `tracert` (Windows) or `traceroute` (Linux/Mac) command to trace the path to a remote server. Document the intermediate hops and their IP addresses.
4. Use the `netstat` command to display active connections, listening ports, and network statistics.
 - i. Document and explain the various parameters and their significance.
 - ii. Use `netstat -r` or `route` to display the routing table of your system. Identify the default gateway and other routes.
5. Use the `arp -a` command to display the ARP table of your system.
 - i. Identify the MAC addresses corresponding to different IP addresses.
 - ii. Clear the ARP cache using `arp-d` and verify the cache is cleared. Re-populate the ARP table by pinging different hosts on the network and verify the entries.
6. Use the `nslookup` command to query the DNS records of a domain (e.g., `google.com`).
 - i. Identify and document the IP addresses associated with the domain.
 - ii. Use the `dig` command (Linux/Mac) for a more detailed DNS query and compare the output with `nslookup`
7. Use the `nmcli` command (Linux) or `netsh wlan show networks` (On Windows) to scan for available Wi-Fi networks and connect to a specified network. Document the steps and verify the connection.
8. Use the `tcpdump` command (Linux) or Wireshark to capture network packets.
 - i. Capture and analyze traffic for a specific protocol (e.g., HTTP) and identify key details like source and destination IPs, ports, and packet content.
 - ii. Filter captured packets to display only traffic to/from a specific IP address or port using `tcpdump` for Wireshark filters.
9. Use the `nmap` command to perform a network scan of your local network.
 - i. Identify active hosts, open ports, and running services.
 - ii. Perform a more detailed scan with service/version detection using `nmap -sV` and analyze the results.
10. Use the `iptables` command (Linux) to set up basic firewall rules. On Windows, use `netsh advfirewall`. Block all incoming traffic except for SSH and HTTP, and verify the rules are working.
11. Use the `route` command to add a static route to a specific network.
 - i. Verify the route using `route -n` (Linux) or `route print` (Windows).
 - ii. Set up IP forwarding on a Linux system using `sysctl` to enable packet forwarding. Test the configuration by pinging through the system acting as a router.

PAPER XII: ALGORITHM DESIGN TECHNIQUES

UNIT-I:

Algorithm specification: Pseudo code, Asymptomatic Analysis, Space complexity and time complexity, Analysis and design of Insertion sort algorithm, Divide and Conquer paradigm, Recurrence relations, Solving Recurrences: Substitution methods, Recursion tree method, and Master method.

UNIT-II:

Searching and Sorting: Analysis of Linear Search, Binary Search, Merge Sort and Quick Sort, Heap Sort. Hashing: Hash functions, Hash table, Collision resolution: Chaining and Open Addressing (Linear probing, Quadratic probing, Double hashing).

UNIT-III:

Greedy Technique: General Method, Applications: Fractional Knapsack Problem, Job Sequencing with Deadlines, Huffman Codes.

Dynamic Programming: General Method, Applications: Matrix Chain Multiplication, longest common subsequence, 0/1 Knapsack.

UNIT-IV:

Graph Algorithms, Topological sort, Minimum Spanning Trees: Prim's and Kruskal's algorithm, Single-source shortest paths: Bellman-Ford algorithm, Dijkstra's algorithm.

Text Book:

- ✓ Introduction to Algorithms, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, PHI.
- ✓ Fundamentals of Computer Algorithms. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, University Press

Reference Book:

Algorithm Design, by Jon Kleinberg, Eva Tardos.

LAB: ALGORITHM DESIGN TECHNIQUES

1. Write C / C++ Program to implement Insertion Sort
2. Write C / C++ Program to implement Merge Sort
3. Write C / C++ Program to implement Quick Sort
4. Write C / C++ Program to implement Heap Sort
5. Write C / C++ Program to implement Hashing
6. Write C / C++ Program to implement Fractional Knapsack
7. Write C / C++ Program to implement Matrix Chain Multiplication
8. Write C / C++ Program to implement Longest Common Subsequence
9. Write C / C++ Program to implement Prim's Algorithm
10. Write C / C++ Program to implement Kruskal's Algorithm
11. Write C / C++ Program to implement Dijkstra's Algorithm

PAPER XIII: WEB TECHNOLOGIES

UNIT-I:

Introduction to the Internet, Internet Protocols, World Wide Web (WWW): Introduction, History, HTTP and HTTP methods, Web Browser, Web Server and their examples, Web page, working principles of WWW. Web Development: Introduction, Front-end and Backend Development Technologies. Concepts of Client-Server communication.

HTML: Introduction, characteristics, basic structure of an HTML document, understanding basic HTML tags and attributes, creating an HTML document. Working with tags for text- formatting, lists, hyperlink, images, tables, frames, multimedia. HTML forms and its basic elements.

UNIT-II:

Cascading Style Sheets (CSS): Introduction, Benefits of using CSS, Understanding the Syntax, CSS Selectors, Using CSS: External, Internal Inline CSS. Comments in CSS. Basic CSS Properties: Color, Background, Text, Font, List, Table, Display.

CSS Box Model: Introduction, working with Margin, Border, and Padding. Pseudo-class & element, working with block elements, Scrolling text, Navigation Bar and Drop Downs.

UNIT-III:

JavaScripts: Introduction, Features, Benefits, Creating Simple JavaScript. Using JavaScript in HTML. Exploring Popup Boxes: alert, confirm, prompt box. Displaying outputs in JavaScript.

Programming using JavaScript: Data types, Variables, Operators, Expressions (Arithmetic, String, Logical), Comments. Control Statements: Conditional, Looping and Jump Statements. Functions (built-in & user defined) and their usage. Working with Array and Date Objects. Introduction to DOM, Event handling and Form validation in JavaScript.

UNIT-IV:

PHP: Features, Print/echo statement, Data Types, Variables, Constants, Strings, Arrays, Operators. Control Structures: Conditional, Looping & Jump Statements. Brief overview of Arrays, Functions: String, Date-Time, Mathematical and User-defined functions.

Embedding PHP in HTML, Reading Form data of a Web Page. Introduction to PHP with Database: Connecting to Database, Creating Table, inserting records, modifying data and retrieving data and displaying in HTML.

Text Book:

Web Technologies (Black Book), DreamTech Press

Reference Books:

- ✓ Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP 4th Edition by Ivan Bayross.
- ✓ HTML, XHTML and CSS Bible, 5ed, Willey India-Steven M. Schafer.

LAB: WEB TECHNOLOGIES

1. Create a Web Page to display “Hello HTML”.
 - Display the same using different headings: h1 to h6
 - Apply bold, italic formats
 - Change text color, background of the page
2. Create a Web Page to display the list of Fruits using both ordered and unordered list. Recreate the lists using images of the same items.
3. Use the web page created in Question Number 2. When the user clicks on the image of a Fruit, it should open a new page that contains an image of the Fruit along with its benefits (use multiple paragraphs, and make sure the image is aligned properly).
4. Create a web page that displays details of the Fruits in Tabular format. Use serial no., name, color, taste, price/kg.
5. Create a Web Page that displays a video file (record/create a video of your own).
 - Apply various controls such as play, pause, volume.
 - Apply autoplay, muted and both.
 - Display a Youtube Video in your Web Page.
6. Design a Student registration form to collect various data about a Student which includes Name, Age, Gender (M, F, O), Mobile No., Email ID, Stream (Science/Arts/Commerce in drop-down), Choice for participating in NCC, NSS, YRC (use check box), and two buttons for reset and submit respectively. Display the form at the center of the page with proper alignment of each item in the form.
7. Use the web page created in Question no. 4. Use CSS and apply various styling to the text, colors to each row of the table, styling to borders and background color of the table.
8. Create a Web page with a Horizontal Navigation bar containing four items such as Home, College, Students, Teachers. The first item should be active, by default. The background color of the item changes when the user moves the mouse over it.
9. Create a Web Page that asks the user to enter the number of Students, and then iteratively ask the details of each Student and display them in a list, using JavaScript.
10. Modify the above program (Q. 9) to display the details in a tabular format, dynamically using JavaScript.
11. Create a Web Page with two text fields and four buttons to perform arithmetic operations such as Addition, Subtraction, Multiplication and Division. The user has to enter numbers in the text fields and press any of the above buttons. The JavaScript program should perform corresponding operations and display the result in the same page. [NOTE: Use different functions for each operation]
12. Use JavaScript and validate the form data in the Student registration Page (created in Q. no. 6). When the user clicks the Submit button, the program has to validate that every field contains valid data. [NOTE: validate name, age, gender, email, mobile number]
13. Write a PHP program that asks the user to enter a number and finds the factorial of it.
14. Write a PHP program that creates a Table in a database with a number of columns as determined by the fields in the Student registration form created above.
15. Write a PHP program to store the data of the Student registration form in a Database.
16. Write a PHP program that asks the user to enter a Student name and display the Student details retrieved from the database in the same page.
17. Students are required to combine the Student Registration form, and PHP program (s) [which interact with Database] to see the dynamic updation of the Student registration data in the Database when a new Student is registered.

SEMESTER-VI

PAPER XIV: ARTIFICIAL INTELLIGENCE

UNIT-I:

Introduction to AI, Scope of AI, Characteristics of AI problems, Turing test, Concept of Intelligent agents, Approaches to AI problem-solving, State space search, production system, Uninformed search: Breadth-First, Depth-First, Iterative deepening, bidirectional and beam search.

UNIT-2:

Informed/Heuristic search: Generate-and-Test, Hill climbing, Best-first search, A* algorithm, Problem reduction, AO*, Constraint satisfaction, Solution of CSP using search, Means-End analysis.

UNIT-3:

Knowledge Representation: Propositional logic and Predicate logic along with their resolution principles, Unification algorithm, forward and backward chaining and conflict resolution, Semantic nets, Frames, Conceptual dependencies, Scripts.

Reasoning under uncertainty: Bayesian Belief networks, Dempster Shafer theory

UNIT-4:

Natural language processing: Introduction, Levels of knowledge in language understanding, , Phases of Natural language understanding, top-down and bottom-up parsing, transition networks.

Expert Systems: Introduction, Architecture, Expert system development cycle, Examples of ES: Mycin and Dendral.

Text Books:

- ✓ Artificial Intelligence - A Modern Approach by Stuart J. Russell & Peter Norvig, Prentice Hall
- ✓ Artificial Intelligence by Rajiv Chopra, S. Chand Pubs.

Reference Books:

- ✓ D.W. Patterson, Introduction to A.I and Expert Systems, PHI Pub.
- ✓ Artificial Intelligence by Rich, Knight, and Nair, McGraw Hill

LAB: ARTIFICIAL INTELLIGENCE

1. Write a Python program to implement Depth-First Search (DFS) for a given graph. Test your program on a graph with at least 5 nodes. Verify your program by printing the order in which nodes are visited.
2. Write a Python program to implement Breadth-First Search (BFS) for a given graph. Use a queue to manage the nodes to be explored. Test your program on a graph with at least 5 nodes and print the order of node visits
3. Write a Python program to implement Uniform Cost Search (UCS) for finding the shortest path in a weighted graph. Test your program on a graph with at least 5 nodes and varying edge weights.
4. Write a Python program to implement the A* search algorithm. Your program should take a graph, a start node, a goal node, and a heuristic program as input. Test your implementation on a grid-based graph where the heuristic is the Manhattan distance.
5. Write a Python program to implement Greedy Best-First Search. Use a heuristic program to guide the search.
6. Write a Python program to solve a maze using the A* search algorithm. Represent the maze as a grid, where 0 indicates an open cell and 1 indicates a wall. Use Manhattan distance as the heuristic.
7. Write a Python program to implement the Minimax algorithm with Alpha-Beta pruning for a simple game (e.g., Tic-Tac-Toe).
8. Write a Python program to implement the Hill Climbing algorithm with random restarts. Test your program on a problem where the solution landscape has multiple peaks.
9. Write a Python program to represent the state of the 8-puzzle. Use a 2D list or a single list with 9 elements to represent the tiles. Implement a program to display the puzzle state.
10. Write a Python program to generate all possible moves (up, down, left, right) from a given state in the 8-puzzle. Ensure that your program checks for the boundaries of the puzzle.

PAPER XV: PROJECT WORK-I

A student has to do a Project work under the guidance of a faculty member. After completing the project, the student has to submit a project report which has to be evaluated by an external examiner. The model template for the project report can be as follows:

1. Title of the project
2. Declaration (by the student)
3. Certificate (of the project guide)
4. Acknowledgement
5. Abstract
[Provide a brief summary of your project, including its objectives, methods, and key findings.]
6. Table of Contents
Introduction
Literature Review
Methodology
Results
Discussion
Conclusion
References
7. Introduction
[Describe the background and context of your project, including the problem statement and objectives.]
8. Literature Review
[Review relevant literature related to your project, discussing previous research, theories, and concepts.]
9. Methodology/
[Explain the methods you used to conduct your research or project, including data collection, analysis techniques, and any tools or software used.]
10. Implementation/Software development
11. Results
[Present the findings of your research or project, using tables, figures, or graphs as needed to illustrate key points.]
12. Discussion
[Interpret your results and discuss their implications, relating them back to your research objectives and the broader context of your field.]
13. Conclusion
[Summarize the main findings of your project and their significance, as well as any recommendations for future research or applications.]
14. References
[List all sources cited in your project using a consistent citation style (e.g., APA, MLA).]

The evaluation pattern of the project will be as follows:

- i. Problem formulation and definition
- ii. Execution of code & results
- iii. Documentation
- iv. Clarity in presentation
- v. Performance in the Viva voce

SEMESTER-VII

PAPER XVI: CYBER SECURITY

UNIT-I:

Introduction: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles. Confidentiality, Integrity, Availability, Computer Criminals, Motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT-II:

Cryptography: History and development of Cryptography. Substitution and affine ciphers, Transpositions Cipher, Confusion, diffusion, Symmetric, Asymmetric Encryption. DES, Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates.

UNIT-III:

Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation-Infected Content - Viruses, Propagation -Vulnerability Exploit-Worms, Propagation-Social Engineering-SPAM E-Mail, Trojans, Payload-System Corruption, Payload-Attack Agent-Zombie, Bots, Payload-Information Theft-Keyloggers, Phishing, Spyware, Payload-Stealth-Backdoors, Rootkits, Countermeasures.

Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of-Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls

UNIT-IV:

Classification of cyber crimes, Common cyber crimes- cyber crime targeting computers and mobiles, cyber crime against women and children, financial frauds, social engineering attacks, malware and ransomware attacks, zero day and zero click attacks, Cybercriminals modus-operandi , Reporting of cyber crimes, Remedial and mitigation measures, Legal perspective of cyber crime, IT Act 2000 and its amendments, Cyber crime and offences, Organisations dealing with Cyber crime and Cyber security in India, Case studies.

Text Books:

- ✓ W. Stallings, L. Brown, Computer Security: Principles and Practice, 4th edition, Pearson Education, 2018.
- ✓ Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley
- ✓ C. P. Pfleeger, S. L. Pfleeger; Security in Computing, Prentice Hall of India, 2006
- ✓ Marc H. Rogers, "Cybersecurity: A Practical Guide to the Law of Cyber Risk", Wiley

PAPER XVII: THEORY OF COMPUTATION

UNIT-I:

Introduction to Finite Automata, Alphabets, Strings, Languages and Problems. FINITE AUTOMATA (FA): Deterministic Finite Automata (DFA)-Formal definition, language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, the extended transition function, the language of an NFA, Equivalence of NFA and DFA, NFA with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of DFA's.

UNIT-II:

REGULAR EXPRESSIONS (RE): Definition of RE, The operators of Regular Expressions, Building RE, Finite Automata and Regular Expressions- Converting DFA's to Regular Expressions, Converting Regular Expressions to Automata, Regular grammars, and FA, proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages. Decision properties of Regular Languages.

UNIT-III:

Context Free Grammars and Languages: Context Free Grammars (CFG): Definition, Derivations using a grammar, trees, Leftmost and Rightmost derivations. Ambiguity in grammars and languages: Removing ambiguity from grammars, Inherent ambiguity. Properties of Context-Free Languages: Normal forms for CFGs; Eliminating useless symbols, Eliminating epsilon productions, Eliminating unit productions, Chomsky Normal Form (CNF), Pumping Lemma for Context Free Languages. Decision properties of CFL's.

UNIT-IV:

Pushdown Automata: Definition, Instantaneous Descriptions of a PDA, The language of a PDA: Acceptance by Final State, Acceptance by empty stack. Equivalence of PDA's and CFG's. Introduction to Turing Machine: Notation, Instantaneous Descriptions for Turing machines, Transition Diagrams, Language, Turing machines and Halting. Universal Turing Machines.

Text Book:

John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory Languages and Computation, Pearson Education, India.

Reference Books:

- ✓ Sipser, M., Introduction to the Theory of Computation, 3rd Edition, Cengage Learning
- ✓ John C Martin, Introduction to languages and the Theory of Computation, McGraw- Hill Publishing.
- ✓ Automata Theory by K.L.P. Mishra and N. Chandrasekaran, PHI Learning Pvt. Ltd

PAPER XVIII: MATHEMATICS FOR MACHINE LEARNING

UNIT-I:

Introduction, Vector Spaces, Linear Independence, Dimensions and Basis, Linear Transformations, Norms and Spaces, Orthogonal Complement and projection, Systems of Linear equations- Gaussian Elimination, LU decomposition

UNIT-II:

Matrix and Operations: Addition, Multiplication, Transpose, Inverse, Rank of a matrix, Matrix factorization: Matrix properties and trace, Eigenvalues and Eigenvectors, Cholesky decomposition, Singular value decomposition, diagonalization.

Gradient, convex sets and convex functions, Properties of convex functions, Gradient of matrices, Numerical Optimization in ML, Gradient descent optimization

UNIT-III:

Applications of Probability, Events and Sample Spaces, Axioms of Probability, Conditional Probability, Baye's Theorem, Random Variable, Expectation, Variance, Concept of Distributions – Bernoulli, Binomial, Poisson, Exponential and Normal.

UNIT-IV:

Mean, Median, Mode, Standard Deviation, Regression - Linear, Multiple and Logistic regression, Least Square Method.

Text Book:

M. P. Deisenorth, A Aldo Faisal, Cheng Soon Ong, "Mathematics for Machine Learning", Cambridge University Press.

Reference Books:

- ✓ Samuel Hack, "Machine Learning Mathematics".
- ✓ Richard Han, "Math for Machine Learning".

PAPER XIX: QUALITY ASSURANCE AND TESTING

UNIT-I

Quality Revolution, Software Quality, Role of Testing, Verification and Validation, Failure, Error, Fault and Defect, Notion of Software Reliability, Objective of Testing, What is a Test Case?, Expected Outcome, Concept of Complete Testing, Testing Activities, Test Oracle, Testing Levels, Regression Testing, White-Box and Black Testing, Test Planning and Design, Monitoring and Measuring Test Execution, Test Tools and Automation

Unit Testing: Concept of Unit Testing, Static and Dynamic unit Testing, Mutation Testing, Debugging, Unit Testing in extreme Programming.

UNIT-II

Control Flow Testing: Outline of Control Flow Testing, Control Flow Graph, Path in a CFG, Path selection Criteria, All-Path Coverage Criterion, Statement Coverage Criterion, Branch Coverage Criterion, Generation of Test Input, Example of Test Data Selection.

Data Flow Testing :Data Flow Anomaly,. Overview of Dynamic Data Flow Testing, Data Flow Graph, Data Flow Terms, Data Flow Testing Criteria, Comparison of Data Flow Test Selection Criteria, Feasible Paths and Test Selection Criteria, Comparison of Testing Techniques.

System Integration Testing :Concept of Integration Testing, Different Types of Interfaces and Interface Errors, Granularity of System Integration Testing, System Integration Techniques, Software and Hardware Integration, Test Plan for System Integration, Off-the-Shelf Component Integration, Off-the-Shelf Component Testing, Built-in Testing

UNIT- III

System Test Categories :Basic Tests, Functionality Tests, Robustness Tests, Interoperability Tests, Performance Tests, Scalability Tests, Stress Tests, Load and Stability Tests, Reliability Tests, Regression Tests, Documentation Tests.

Functional Testing :Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing, Error Guessing, Category Partition.

System Test Planning And Automation :Structure of a System Test Plan, Introduction and Feature Description, Assumptions, Test Approach, Test Suite Structure, Test Environment, Test Execution Strategy, Test Effort Estimation, Scheduling and Test Milestones, System Test Automation, Evaluation and Selection of Test Automation Tools, Test Selection Guidelines for Automation, Characteristics of Automated Test Cases, Structure of an Automated Test Case, Test Automation Infrastructure.

Acceptance Testing :Types of Acceptance Testing, Acceptance Criteria, Selection of Acceptance Criteria, Acceptance Test Plan, Acceptance Test Execution, Acceptance Test Report, Acceptance Testing in extreme Programming.

UNIT-IV

Software Reliability: Definition, Factors Influencing Software Reliability, Application of Software Reliability, Operational Profiles.

Software Quality :Five Views of Software Quality, McCall's Quality Factors and Criteria, Quality Factors Quality Criteria, Relationship between Quality Factors and Criteria, Quality Metrics, ISO 9126 Quality Characteristics, ISO 9000:2000 Software Quality Standard ISO 9000:2000 Fundamentals, ISO 9001:2000 Requirements.

Maturity Models: Basic Idea in Software Process, Capability Model(CMM) Model, Architecture, Five Levels of Maturity and Key Process Areas, Common Features of Key Practices, Application of CMM, CMMI, Test Process Improvement (TPI), Testing Maturity Model (TMM).

Textbook:

- Software Quality Assurance, Daniel Galin, Pearson Education

Reference Book:

- Software Testing Laboratory Manual by Rahul A. Palakar, Computer Science Dept. B.T.Patil & Sons Polytechnic, Available in the Web.

SEMESTER VIII

PAPER XX: INTRODUCTION TO DATA SCIENCE

UNIT-I:

Data Scientist's Tool Box: Turning data into actionable knowledge, introduction to the tools that are used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.

UNIT-II:

R Programming Basics: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scope rules, dates and times, Loop functions, debugging tools, Simulation, code profiling.

UNIT-III:

Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and other sources in various formats, basics of data cleaning and making data "tidy".

UNIT-IV:

Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data.

Text Book:

Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Front line" Schroff / O'Reilly, 2013.

Reference Books:

- ✓ Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking by O'Reilly, 2013.
- ✓ John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
- ✓ Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1st Edition, by Wiley, 2013.

LAB: INTRODUCTION TO DATA SCIENCE

1. Study of basic Syntaxes in R
2. Implementation of vector data objects operations
3. Implementation of matrix, array and factors and perform variance analog in R
4. Implementation and use of data frames in R
5. Create Sample (Dummy) Data in R and perform data manipulation with R
6. Study and implementation of various control structures in R
7. Data Manipulation with dplyr package
8. Data Manipulation with data.table package
9. Study and implementation of Data Visualization with ggplot2
10. Study and implementation data transpose operations in R

PAPER XXI: INTERNET OF THINGS

UNIT-I:

Emergence of IoT: Introduction, Evolution of IoT, Layered Architecture, Networking Components, Addressing Strategies. IoT Enabling Technologies: Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Embedded Systems.

IoT Levels and Deployment Templates: Introduction, IoT Level-1 to Level-6. IoT Sensing and Actuation: Introduction, Sensors, their characteristics and types. Actuators, their characteristics and types.

UNIT-II:

Application Domains of IoT: Introduction, Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Lifestyle. Paradigms, Challenges and Future: Evolution of new IoT Paradigms, Challenges associated with IoT, Emerging Pillars of IoT.

IoT Processing Topologies: Data Format, importance of processing, processing topologies, IoT device design and selection considerations, Processing offloading.

UNIT-III:

IoT Connectivity Technologies: Introduction & types, IEEE 802.15.4, Zigbee, Thread, Wireless HART, RFID, NFC, Z-Wave, LoRa, Wi-Fi, Bluetooth.

IoT Communication Technologies: Introduction, Protocols for Infrastructure, Discovery, Data, Identification, Device Management and Semantic.

UNIT-IV:

Introduction to various IoT Development Platforms, Sensor development boards/platforms, Arduino versus Raspberry Pi. Arduino: Introduction, installation & setup. Introduction to Sketch, Data Types, Operators, Control Statements, Arrays, String, Common Functions and Libraries.

Text Books:

- ✓ Introduction to IoT by S Mishra, A. Mukharjee & A. Roy, Cambridge University Press.
- ✓ Internet of Things: A hands-on approach by A. Bahga & V. Madisetti, University Press.

Reference Books:

- ✓ "The Internet of Things: Enabling Technologies, Platforms, and Use Cases" by Pethuru Raj and Anupama C. Raman, CRC Press
- ✓ D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry; IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, PHI.

PAPER XXII: MACHINE LEARNING

UNIT-I:

Introduction: Brief Introduction to Machine Learning Supervised Learning Unsupervised Learning Reinforcement Learning, Overview of supervised learning, classification, and regression problems, Statistical Pattern Recognition: Bayes Decision Theory, Minimum Error and Minimum Risk Classifiers, Discriminant Function and Decision Boundary, Normal Density, Discriminant Function for Discrete Features. Naïve Bayes Classification.

UNIT-II:

Classification: K-nearest neighbourhood (KNN) classifier, variation of k-NN classifiers. Decision tree learning, Issues in Decision tree learning. Artificial Neural Network: Introduction – Fundamental concept – Evolution of Neural Networks – Basic Models of Artificial Neural Networks – Important Terminologies of ANNs – McCulloch-Pitts Neuron – Linear Separability – Back-Propagation Network – Radial Basis Function Network, multi-level classification.

UNIT-III:

Model Assessment and Selection: Loss function, test and training error, Bias, Variance, and model complexity, Bias-variance trade off, Bayesian approach and BIC, Cross-validation, Boot strap methods, Performance of Classification algorithms (Confusion Matrix, Precision, Recall and ROC Curve), Introduction to Gradient Descent, The Curse of Dimensionality, Dimensionality Reduction, Principal Component Analysis.

UNIT-IV:

Unsupervised Learning and Random forests: Cluster analysis (k-means, Hierarchical clustering, DBSCAN, spectral clustering), Gaussian mixtures and EM algorithm, Random forests and analysis. Introduction to Deep Learning, Case studies on Digit classification and Image recognition.

Text Books:

- ✓ Tom M. Mitchell, Machine Learning, McGraw Hill Education, Indian Edition.
- ✓ Alpaydin, E., Introduction to Machine Learning. United Kingdom: MIT Press.
- ✓ S.N. Sivanandan and S.N. Deepa, Principles of Soft Computing, Wiley India.

Reference Books:

- ✓ C. M. Bishop –Pattern Recognition and Machine Learning, Springer.
- ✓ Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning-Data Mining, Inference, and Prediction, Second Edition, Springer Verlag.

LAB: MACHINE LEARNING

1. Write Python programs for Creation and Loading different types of datasets in Python using the required libraries.
 - a) Creation using pandas
 - b) Loading CSV dataset files using Pandas
 - c) Loading datasets using sklearn
2. Write a python program to compute Mean, Median, Mode, Variance, Standard Deviation using Datasets
3. Demonstrate various data pre-processing techniques for a given dataset. Write a python program to compute
 - a) Reshaping the data,
 - b) Filtering the data,
 - c) Merging the data
 - d) Handling the missing values in datasets
 - e) Feature Normalization: Min-max normalization
4. Implement Naïve Bayes theorem to classify the English text using Python.
5. Write a python program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Implement linear regression using python.
7. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Vary the number of k values as follows and compare the results:
 - a. 1
 - b. 3
 - c. 5
 - d. 7
 - e. 11
8. Build Artificial Neural Network model with back propagation on a given dataset.
9. Implement Dimensionality reduction using Principal Component Analysis (PCA) method on a dataset (For example Iris).
10. Write a python program to implement K-Means clustering Algorithm.
11. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering.
12. Write a python program to implement DBSCAN clustering Algorithm.
13. Consider different classification problem like Digit Classification, Image Classification. Implement that using different ML techniques and compare their result and make a report.

PAPER XXIII: PROJECT WORK-II

Student of four-year Bachelor's Degree (Honours without Research) are required to take up a software development project under the guidance of a faculty Member. The students are expected to initiate the project work during seventh semester and complete the Project in the Eighth Semester and submit a project report for evaluation by an external examiner. The template for project report can be as Paper XV: Project Work-I.

The evaluation pattern of the project will be as follows:

- i. Problem formulation and definition
- ii. Execution of code & results
- iii. Documentation
- iv. Clarity in presentation
- v. Performance in the Viva voce

Research Methodology & Dissertation

(For 4-Year Hons. with Research)

(Research Methodology: 4 credits + Dissertation: 8 credits)

Student of four-year Bachelor's Degree (Honours with Research) are required to take up Research Projects under the guidance of a faculty Member. The students are expected to initiate the project work during seventh semester and complete the Research Project in the Eighth Semester. The Research outcome of their project work may be published in peer-reviewed journals or may be presented in conferences/ seminars or may be patented. The evaluation of the dissertation has to be made by an external examiner.

RESEARCH METHODOLOGY

UNIT-I:

Research Methodology: An Introduction: Meaning and objective of Research Methodology, Motivation in research, types of research, Research Approaches, significance of research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, Problems Encountered by Researchers in India. Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, Illustration and Conclusion

UNIT-II:

Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs and Conclusion. Methods of Data Collection: Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires, Collection of Data through Schedules, Difference between Questionnaires and Schedules, Some Other Methods of Data Collection, Collection of Secondary Data, Selection of Appropriate Method for Data Collection.

UNIT-III:

Processing and Analysis of Data: Processing Operations, Some Problems in Processing, Elements/Types of Analysis, Statistics in Research, Measures of Central Tendency, Measures of Dispersion, Measures of Asymmetry, Measures of Relationship, Simple Regression Analysis, Multiple Correlation and Regression, Curve Fitting. Testing of Hypotheses: What is a Hypothesis, Basic Concepts Concerning Testing of Hypotheses, Procedure for Hypothesis Testing, Important Parametric Tests, Hypothesis Testing of Means, Hypothesis Testing for Comparing Two Related Samples, Hypothesis Testing of Proportions, Hypothesis Testing for Difference between Proportions.

UNIT-IV:

Chi-square Test: Chi-square as a Test for Comparing Variance, Chi-square as a Non- parametric Test, Conditions for the Application of χ^2 Test, Steps Involved in Applying Chi- square Test, Alternative Formula, Yates 'Correction, Conversion of χ^2 into Phi Coefficient, Conversion of χ^2 into Coefficient by Contingency, Important Characteristics of χ^2 Test. Analysis of Variance: Analysis of Variance (ANOVA), The Basic Principle of ANOVA, ANOVA Technique

Text Books:

- ✓ Research Methodology Methods and Techniques, Kothari, C. R., New Age.
- ✓ Research Methodology: a step-by-step guide for beginners, Kumar, Pearson Education.
- ✓ Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd.
- ✓ Research Methodology, Sharma, N. K., KSK Publishers, NewDelhi.