NEP

Syllabus for

B.Sc. Computer Science

Major,

Minor

and

Interdisciplinary

Sem			Major	Lab		Minor	Lab		Multidisciplinary	Lab
I	1	CMSDSC101T	Computer Fundamentals and programming with C	Y	CMSMIN101T	Computer Fundamentals and programming with C	Y	CMSCOR101T	Computer Fundamentals and programming with C	Y
		CMSDSC101P	Computer Fundamentals and programming with C		CMSMIN101P	Computer Fundamentals and programming with C		CMSCOR101P	Computer Fundamentals and programming with C	
II	2	CMSDSC202T	Data Structures using C++	Y	CMSMIN202T	Data Structures using C++	Y	CMSCOR202T	Data Structures using C++	Y
		CMSDSC202P	Data Structures using C++		CMSMIN202P	Data Structures using C++		CMSCOR202P	Data Structures using C++	
III	3	CMSDSC303T	Basic Electronics and Digital Logic	Y	CMSMIN303T	Operating Systems	Y	CMSCOR303T	Operating Systems	Y
		CMSDSC303P	Basic Electronics and Digital Logic		CMSMIN303P	Operating Systems		CMSCOR303P	Operating Systems	
IV	4	CMSDSC404T	Object Oriented Programing	Y				CMSCOR404T	Database Management System	Y
		CMSDSC404P	Object Oriented Programing							
	5	CMSDSC405T	Operating Systems	Y						
		CMSDSC405P	Operating Systems							
	6	CMSDSC406T	Computational Mathematics	N				CMSCOR404P	Database Management System	
	7	CMSDSC407T	Computer System Architecture	N						
V	8	CMSDSC508T	Analysis of Algorithms	Y				CMSCOR505T	Computer System Architecture	N
		CMSDSC508P	Analysis of Algorithms							
	9	CMSDSC509T	Database Management System	Y						
		CMSDSC509P	Database Management System							
	10	CMSDSC510T	Software Engineering	Y						
		CMSDSC510P	Software Engineering							
	11	CMSDSC511T	Networking	N						
VI	12	CMSDSC612T	Graphics and Multimedia	Y				CMSCOR606T	Computer Networks	N
		CMSDSC612P	Graphics and Multimedia							
	13	CMSDSC613T	Artificial Intelligence	Y						
		CMSDSC613P	Artificial Intelligence							
	14	CMSDSC614T	Microprocessor and VLSI Design	Y						
		CMSDSC614P	Microprocessor and VLSI Design							
	15	CMSDSC615T	Theory of Computation	N						
VII	16	CMSDSC716T	Image Processing and Computer Vision	Y	CMSSMC701T	Database Management System	Y			
		CMSDSC716P	Image Processing and Computer Vision		CMSSMC701P	Database Management System				
	17	CMSDSC717T	Compiler Design	N						
VIII	18	CMSDSC818T	Machine Learning	Y						
		CMSDSC818P	Machine Learning							
	19	CMSDSC819T	Cyber Security and IoT	N						
	20	CMSDSC820T	Big Data	N						
	21	CMSDSC821P	Application Design & Development	LAB						

VIII	18	CMSDSC818T	Machine Learning	Y	Honors with Research
		CMSDSC818P	Machine Learning		
	19	CMSRES801P	Major Project	LAB	

	MDC	Lab		SEC	Lab
CMSHMD101T / CMSGMD101T	Computer Fundamentals	N	CMSHSE101M / CMSGSE101M	R Programming	Y
CMSHMD201T / CMSGMD201T	Computer Fundamentals	N	CMSHSE202M / CMSGSE202M	Python Programming	Y
CMSHMD301T / CMSGMD301T	Computer Fundamentals	N			

CORE COURSES (MAJOR / HONOURS IN COMPUTER SCIENCE)

CMSDSC101T/CMSMIN101T/CMSCOR101T:
Computer Fundamentals and programming with C Theory: 45 Lectures

1. Overview of C (5 Lectures)

History, Basic Structure, Algorithms, Structured programming constructs. Character sets, Tokens, Keywords, Constants, Variables, Data Types, Declaration of storage classes.

2. Operators, Expressions and Preprocessor (8 Lectures)

Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional, Bitwise, Special operator, Operator Precedence and Associativity; Arithmetic Expressions, Evaluation of expression, type casting. Comments, Input and output operations. Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

3. Decision and Loop Control Structure (7 Lectures)

If-else statements, Nested if-else, switch, Conditional operator. While, do-While, for loop, break statements, continue statements, goto statements.

4. Functions and Arrays (7 Lectures)

Utility of functions, Call by Value, Call by Reference, Functions returning value, 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.

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), Introduction to Multi-dimensional arrays, return statement, return values and their types, String handling with arrays, String handling functions, recursion

5. User defined Data types and Memory Allocation (6 Lectures)

Enumerated data types, Structures. Structure arrays, Pointers to Functions and Structures, Unions. 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

6. Pointers & File Access (12 Lectures)

Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation. Opening and closing a file (use of fstream header file, ifstream, ofstream), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files,

Text Books

1. Let Us C, Kanetkar, BPB Publication.
2. Programming in ANSI C, Balaguruswamy, McGraw Hill.

Reference Books

1. Programming with C, Byron S. Gottfried, McGraw Hill.
2. The C Programming Language, Kernighan and Dennis Ritchie, PHI.
3. The Complete reference C, Herbert Schildt, McGraw Hill.
4. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

CMSDSC101P/CMSMIN101P/CMSCOR101P:
Computer Fundamentals and programming with C Practical: 60 Lectures

Instruction: Use an open source C compiler.

1. Write a program (WAP) to print the sum and product of digits of an integer.
2. WAP to reverse a non-negative integer.
3. WAP to compute the sum of the first n terms of the following series,
   S =1-2+3- 4+5…………….
4. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by the user is Palindrome or not.
6. WAP to compute the factors of a given number.
7. WAP to swap two numbers using macro.
8. WAP to print a triangle of stars as follows (take number of lines from user):

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

10. WAP to perform following actions on an array entered by the user :
    a. Print the even-valued elements
    b. Print the odd-valued elements
    c. Calculate and print the sum and average of the elements of array
    d. Print the maximum and minimum element of array
    e. Remove the duplicates from the array
    f. 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 the array and to quit the program.)
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function passes the address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input 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.
15. Write a program to find the sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operators.
16. Write a menu driven program to perform following operations on strings:
    a. Show address of each character in string
    b. Concatenate two strings without using strcat function.
    c. Concatenate two strings using strcat function.
    d. Compare two strings
    e. Calculate length of the string (use pointers)
    f. Convert all lowercase characters to uppercase

g. Convert all uppercase characters to lowercase
h. Calculate number of vowels
i. Reverse the string

17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration.
19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from the user and store them in an array. Implement an array using pointers. The program will print the array elements in ascending and descending order.
25. Add two distances in a meter kilometer system using structure.
26. Add two complex numbers using structures.
27. Calculate the difference between two time periods using structures.

CMSDSC202T/CMSMIN202T/CMSCOR202T:
Data Structures using C++ **Theory: 45 Lectures**

1. Introduction **(5 Lectures)**
Data Object, Abstract Data Type, Data Structures and Data Types. Types of Data Structures – Linear and non-linear Data Structures. Single and Multi-dimensional Arrays, Address Calculations, Sparse Matrices (Array Representation).

2. Linked Lists **(7 Lectures)**
Singly, Doubly and Circular Lists (Array and Linked representation); Operations on Lists. Sparse Matrices (Linked Representation).

3. Stacks and Queues **(9 Lectures)**
Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack. Array and Linked representation of Queue, De-queue, Priority Queues

4. Recursion **(5 lectures)**
Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

5. Binary Trees **(10 Lectures)**
Introduction; Properties, Binary Trees Traversals (Recursive and Non-Recursive), Binary Search Trees (Insertion, Deletion), Recursive and Iterative Traversals in Binary Search Trees; Threaded Binary Trees (Concept only); Height-Balanced Trees (Concept only).

6. Searching, Sorting and Hashing **(9 Lectures)**
Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Bubble Sort, Comparison of Sorting Techniques. Introduction to Hashing, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining and simple examples.

Text Books

1. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++:, Second edition, PHI, 2009.
2. Sartaj Sahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.

Reference Books

1. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson,1999.
2. D. S. Malik, Data Structure using C++, Second edition, Cengage Learning, 2010.
3. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
4. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using Java", 2003.
5. Samanta, D. "Classic data structures.", PHI, *Terminology* 2 (2001): 1.
6. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.

CMSDSC202P/CMSMIN202P/CMSCOR202P:

Data Structures using C++ Practical: 60 Lectures

Instruction: Use an open source C++ compiler.

1. Write a program to search an element from a list. Give users the option to perform Linear or Binary search.
2. WAP to sort a list of elements. Give users the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List. 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. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List. 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.
8. Perform Queues operations using Circular Array implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using a 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 two numbers (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree:
    - i. Insertion (Recursive and Iterative Implementation)
    - ii. Deletion by copying
    - iii. Deletion by Merging
    - iv. Search a no. in BST
    - v. Display its preorder, postorder and inorder traversals Recursively
    - vi. Display its preorder, postorder and inorder traversals Iteratively

vii. Display its level-by-level traversals
viii. Count the non-leaf nodes and leaf nodes
ix. Display height of tree
x. Create a mirror image of tree
xi. 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 a one-dimensional array.
19. WAP to implement Lower Triangular Matrix using a one-dimensional array.
20. WAP to implement the Upper Triangular Matrix using a one-dimensional array.
21. WAP to implement Symmetric Matrix using a one-dimensional array.

CMSDSC303T: Basic Electronics with Digital Logic Theory: 45 Lectures

1. Elementary Physics of semiconductors (6 Lectures)

Intrinsic and Extrinsic semiconductors, P & N type, Diode & its applications: P-N Junction diodes, Biasing of a junction diode, Depletion region & its effect, Diode as a rectifier, Types of diodes, LED, LCD. Elementary concepts of FET, MOS, CMOS, and realization of TTL circuit

2. Principle of junction transistors (7 Lectures)

Current components of a transistor, Modes of a transistor (CB, CE and CC) and their properties, I/O characteristics of a transistor in CE mode. Relation between parameters of Transistor, Biasing of a transistor: Q point, load line, Self-bias, fixed bias & collector to base bias. Amplifiers: Concepts, Brief idea of Class A & B. Inverters using Transistors–transfer characteristics and threshold voltages.

3. Combinational Circuits (8 Lectures)

Realization of AND and OR Gates using diodes and NOT Gate using transistors, Standard Gate Assemblies, IC chips packaging nomenclature, Half and Full Adder(3 & bit), Multi-bit adders – Ripple carry and Carry Look Ahead Adder, Adder/Subtractor, BCD-Adder.

4. Data selectors/multiplexers: (8 Lectures)

Expansions, reductions, function realization, universal function realization, multi-function realization, Decoders: function realization, Demultiplexer and function realization, Encoder, Priority Encoder, Parity bit Generator/checker, Gray Code Generator, Code Converters.

5. Sequential Circuits (12 Lectures)

Model of Sequential computing, Difference between Combinational and Sequential circuit, RS Latch: using NAND and NOR Gates, RS Latch as a Static RAM Cell, Problems of Basic Latch circuits, Digital Clock – Duty Cycle, Rising time, Falling time, Clocked Flip Flops - SR, JK, D, T, Level Trigger and Edge Trigger, Excitation Functions of each flip-flops, Flip-flops with Preset and Clear, Application of Flip-flops: Asynchronous Counter(UP/DOWN) upto 4 bit counter, Decade Counter, Mod – n Counter, Synchronous Counters – different mod counters, Ring counter, Registers, Shift Registers.

6. Data Converter (4 Lectures)

D/A Conversion principle using basic circuit, R-2R Ladder circuit, Counter based A/D converter, Successive approximation method for A/D conversion.

Text Books

1. Introduction to Computer Science by P. K. Sinha, PHI

2. Electronics Fundamentals and Applications by D. Chattopadhay and P. C. Rakshit, 6th Edition, New Age International (P)

Reference Books

1. Digital Systems Principles and Applications by Ronal J. Tocci and Neal S. Widmer, 8th Edition, PHI
2. Digital Fundamentals by Floyd, Pearson Education
3. Electronics Devices and Circuit Theory by Boylestad, Nashelsky, PHI
4. Digital Circuits and Design by S. Salivahanan, 5th Edition, Oxford University Press
5. Digital Logic and Computer Design by Morris Mano, PHI
6. Digital Principle and Applications by Malvino & Leach, TMH
7. Computer System Architecture by M. Morris Mano

CMSDSC303P: Basic Electronics with Digital Logic Practical: 60 Lectures

1. Implement Half Adder/Half Subtractor/Full Adder/Full Subtractor using Logic Gates. Realize a logic function using basic/universal gates in SOP and POS form. Study the functionalities of 7483 and design a BCD adder using 7483 or equivalent.
2. Design of two level AND – OR, NAND –NAND, NOR-NOR circuits to realize any truth table. Realize XOR in two level and multilevel.
3. Design a 4 bit 2's complement adder – subtractor unit using 7483 or equivalent and XOR gates.
4. Design a circuit to convert BCD numbers to corresponding gray codes.
5. Design a 4:1 MUX using NAND gates. Study of 74153 and 74151. Design Full Adder/Subtractor using MUX.
6. Design a 2:4 decoder using NAND gates. Study of 74155 and 74138. Design Full Adder/Subtractor using decoders.
7. Design a parity generator/checker using basic gates.
8. Design magnitude comparator using basic/universal gates. Study of 7485.
9. Design a seven segment display unit.
10. Realize S-R, D, J-K and T flip-flops using basic gates. (Study the undefined state in S-R flip-flop).
11. Design a shift register (shift left and shift right) using flip-flops. (Study the functional characteristics of IC74194 with emphasis on timing diagram).
12. Design Asynchronous and Synchronous counters. Study of IC 74193.

CMSDSC404T:Object Oriented Programing Theory: 45 Lectures

1. Object-Oriented Programming Overview (4 Lectures)

Principles of Object-Oriented Programming, Data Abstraction and Information Hiding, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

2. Introduction to Java (4 Lectures)

Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, 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),

3. Arrays, Strings and I/O **(8 Lectures)**

Creating & Using Arrays (One Dimensional and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects,

Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System. out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

4. Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata **(14 lectures)**

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

5. Exception Handling, Threading and Database Connectivity **(15 Lectures)**

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.. Accessing and manipulating databases using JDBC.

Text Books

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley"The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.

Reference Books

1. Joshua Bloch, "Effective Java" 2nd Edition, Publisher: Addison-Wesley, 2008.
2. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 19th Edition, Printice Hall.2012
3. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features", 9th Edition, Printice Hall.2013
4. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
5. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
6. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011.
7. "Head First Java ", Orielly Media Inc. 2nd Edition, 2005.
8. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
9. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

CMSDSC404P: Object Oriented Programing Practical: 60 Lectures

Instruction: Use an open source JAVA compiler.

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 learn use of a single dimensional array by defining the array dynamically.
4. To learn use of .lenth in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. 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 Write a program that show working of different functions of String and StringBuffer classes like setCharAt, setLength(), append(), insert(), concat()and equals().

8. 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
9. Modify the ―distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign a second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
10. 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)
11. 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
12. Write a program to show the use of static functions and to pass variable length arguments in a function.
13. Write a program to demonstrate the concept of boxing and unboxing.
14. 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).
15. 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.
16. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
17. Write a program ―Divide By Zero 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.
18. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
19. Write a program to create your own exception types to handle situations specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
20. Write a program to demonstrate priorities among multiple threads.
21. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).

CMSDSC405T/CMSMIN303T/CMSCOR303T:
Operating Systems Theory: 45 Lectures

1. Introduction (7 Lectures)

Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

2. Operating System Organization (6 Lectures)

Processor and user modes, kernels, system calls and system programs.

3. Process Management (16 Lectures)

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks.

4. Memory Management (8 Lectures)

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory

5. File and I/O Management **(8 Lectures)**

Directory structure, file operations, file allocation methods, device management.

Text Books

1. Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A. S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.

Reference Books

1. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
2. W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India. 2008.
3. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

CMSDSC405P/CMSMIN303P/CMSCOR303P:
Operating Systems Practical: 60 Lectures

Note: In Practical Examination 60% weighted in Group I and rest in Group II are allotted.

Group I: C Programs (Use an open source C compiler.)

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
   a. Same program, same code.
   b. Same program, different code.
   c. Before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
5. WRITE A PROGRAM to copy files using system calls.
6. Write a program to implement FCFS scheduling algorithm.
7. Write a program to implement the Round Robin scheduling algorithm.
8. Write a program to implement the SJF scheduling algorithm.
9. Write a program to implement a non-preemptive priority based scheduling algorithm.
10. Write a program to implement a preemptive priority based scheduling algorithm.

Group II: Shell Programs (Use an open source Linux compiler.)

1. Write a Shell Script to determine all prime numbers between two positive integers a and b. a and b are supplied as command line arguments.
2. Write a Shell Script that counts the number of words in a line of text. The line text will be provided as command line arguments.
3. Write a Shell Script/Program to check whether a given string is a Palindrome or not.
4. Write a Shell Script/Program to print the command line arguments in reverse order.
5. Write a Shell Script/Program to modify the cal command of UNIX to include the following.
   i) Print the current month and current year when called without parameter.
   ii) Print the calendar of the month of the current year if only one argument between 1 and 12.

CMSDSC406T: Computational Mathematics Theory: 60 Lectures

Additional Tutorial: 15 lectures

1. **Introduction: (15 Lectures)**

   Sets - Finite and Infinite sets, unaccountably Infinite Sets; functions; Relations, Properties of Binary Relations, Closure, Partial Ordering Relations; Counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.

2. **Growth of Functions: (4 Lectures)**

   Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals Asymptotic Notations (Big O, Big Ω, Big Θ), Summation formulas and properties, Bounding Summations

3. **Recurrence Relations**: **(10 Lectures)**

   Recurrence Relations, Generating Functions, Formulation of Recurrence Relations, Linear Recurrence Relations with constant coefficients and their solutions: Substitution method, Characteristic Equation method, Generating Functions method, Recurrence Trees.

4. **Graph Theory: (12 Lectures)**

   Basic Terminology, Models and Types, multigraphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring.

5. **Probability and Statistics: (15 Lectures)**

   Random experiments, Event space, Random variables, Probability, Conditional probability, Independent events. Probability distribution: Binomial, Poisson, Normal, Joint probability distribution. Statistical Data, Frequency distribution. Measures of Central tendency: Arithmetic mean, Geometric mean, Harmonic mean. Median, Order Statistics, Mode. Measures of dispersion: Range, Quartile deviation, Mean deviation, Standard deviation. Hypothesis and estimation.

6. **Propositional Logic: (4 Lectures)**

   Logical Connectives, Well-formed Formulas, Tautologies, Equivalences.

Text Books

1. C. L. Liu and D. P. Mohapatra, Elements of Discrete Mathematics, Third Edition, Tata McGraw Hill , 2008.
2. K. Rosen, Discrete Mathematics and Its Applications, Eighth Edition, McGraw Hill, 2021.

Recommended Books:

1. T. H. Cormen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, 4th Edition The MIT Press, 2022
2. N. Chandrasekaran and M. Umaparvathi, Discrete Mathematics, Third Edition, PHI Learning Pvt. Ltd., 2022.
3. S. Lipschutz and M. Lipson, Discrete Mathematics (Schaum’s Outlines), Fourth Edition, McGraw Hill, 2021.
4. S. Lipschutz and M. Lipson, Probability (Schaum’s Outlines), Second Edition, McGraw Hill, 2011.
5. M. Spiegel, J. Schiller, R. A. Srinivasan, and D. Goswami, Probability and Statistics (Schaum’s Outlines),

Third Edition, McGraw Hill, 2017.

6. N. G. Das, Statistical Methods (Vol 1 & 2), Tata McGraw Hill, New Delhi, 2010
7. P. K. Giri and J. Banerjee, Statistical tools and techniques, Seventh Edition, Academic Publishers, Kolkata, 2017

CMSDSC407T/CMSCOR505T:
Computer System Architecture Theory: 60 Lectures
Additional Tutorial: 15 lectures

1. Introduction to Computer Architecture: (04 lectures)

Basic Component of Computer Architecture, Von-Neumann Architecture, Flynn's Classification of Computers**.**

2. Data Representation and Basic Computer Arithmetic (12 lectures)

Fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, Multiplication: Shift and Add, Booth Multiplier, Carry Save Multiplier, Division:
Non restoring, Restoring.
Weighted and Non - Weighted Codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Gray Codes, Alphanumeric codes, ASCII, EBCDIC, Conversion of bases, Parity bits. Single bit error detection and correcting codes: Hamming Code.

3. Basic Computer Organization and Design (10 lectures)

Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input -output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.

4. Central Processing Unit (12 lectures)

Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.

5. Memory Organization (12 lectures)

Hierarchical memory organization, Cache memory, Associative memory, mapping, Virtual Memory and Machine, Performance Metrics (like Latency, throughput).

6. Input-Output Organization (10 lectures)

Input / Output: External Devices, I/O modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Chanel.

Text Books

1. Computer System Architecture, M. Mano, Pearson Education,1992
2. Digital Circuits, Vol - I & II, D. Ray Chaudhuri, Platinum Publishers.

Recommended Books:

1. Digital Circuits and Design, S. Salivahanan and S. Arivazhagan, 5 ed., Oxford
2. Fundamentals of Digital Circuits, A. Anand Kumar, 2nd Ed., PHI
3. Computer Architecture and Organization, John P. Hayes, 2nd Ed., Mc Graw Hill
4. Digital Electronics, P. Raja, 3rd Ed., SCITEC

CMSDSC508T: Analysis of Algorithms Practical:45 Lectures

1. Introduction & Analysis Technique (8 Lectures)

Definition, Characteristics, Correctness of Algorithm, Recursive and Non-recursive algorithms. Space and Time Complexity, Efficiency of an algorithm, Growth of Functions, Polynomial and Exponential Complexity, Asymptotic Notations: Big O Notation, Big Ω Notations, Big Θ Notations, Properties: Best case/worst case/average case analysis of well-known algorithms.

2. Searching and Sorting Techniques (5 Lectures)

Elementary Searching (Linear and Binary) and sorting techniques–Bubble Sort, Insertion Sort, Selection Sort, Advanced Sorting techniques - Merge Sort, Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort; Complexity Analysis.

3. Algorithm Design Techniques (12 Lectures)

Concepts and simple case studies of Greedy algorithms. Divide and conquer: Basic concepts, Case study of selected searching and sorting problems as divide and conquer techniques, Dynamic programming: General issues in Dynamic Programming, Case study of Binomial Coefficient computation. Backtracking Algorithms, Case study of N-Queen problem.

3. Graph Representation and Algorithm: (15 Lectures)

Graph traversal algorithms: Breadth First Search, Depth First Search and its Applications, Minimal spanning trees: Prim's Algorithm, Kruskal's Algorithm, Shortest path algorithms: Floyd's Algorithm, Floyd-Warshall Algorithm, Dijkstra's Algorithm, Bellman-Ford algorithm, Graph Coloring Algorithms. Application of Tree, Decision Trees, Red-Black Trees.

5. Classification of Problems: (5 Lectures)

P, NP, Satisfiability, Cook’s Theorem (Statement Only)

Text Books

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sara Basse & A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999

Reference Books

1. Fundamentals of Computer Algorithms, E.Horowitz and Sahani, Galgoti
2. Algorithm Design, Jon Kleiberg and Eva Tardos, Pearson Education
3. Data Structures and Algorithms - K.Mehlhorn , EATCS, Vol. I & Vol. 2

CMSDSC508P: Design and Analysis of Algorithms Practical:60 Lectures

Instruction: Use an open source C++ compiler. The program should report the number of comparisons.

1. Write a program to implement Linear Search.
2. Write a program to implement Binary Search.
3. Write a program to implement Insertion Sort.
4. Write a program to implement Bubble Sort.
5. Write a program to implement Selection Sort.
6. Write a program to implement Merge Sort.
7. Write a program to implement Heap Sort.
8. Write a program to implement Quick Sort .
9. Create a Red-Black Tree and perform following operations on it:
   - i. Insert a node
   - ii. Delete a node
   - iii. Search for a number & also report the color of the node containing this number
10. Write a program to implement the Knapsack problem.
11. Write a program to implement the 4-Queen problem.
12. Write a program to implement Breadth-First Search in a graph
13. Write a program to implement Depth-First Search in a graph.
14. Write a program to implement Prim's Algorithm in a graph.
15. Write a program to implement Kruskal Algorithm in a graph.

CMSDSC509T/CMSMIN707T/CMSCOR404T:

Database Management Systems Theory: 45 Lectures

1. Introduction (4 Lectures)

Characteristics of database approach, data models, database system architecture and data independence.

2. **Entity Relationship(ER) Modeling (5 Lectures)**

Entity types, relationships, constraints.

3. **Relation data model (15 Lectures)**

Relational model concepts, relational constraints, relational algebra, SQL queries.

4. **Database design (12 Lectures)**

Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms (up to BCNF).

5. Transaction Processing (3 Lectures)

ACID properties, concurrency control.

6. **File Structure and Indexing (6 Lectures)**

Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files(Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Text Books

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.

Books Recommended:

1. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGraw Hill,

2010.

2. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

CMSDSC509P/CMSMIN707P/CMSCOR404P:
Database Management Systems Practical: 60 Lectures

Instruction: Use an open source MySQL compiler.

Create and use the following database 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 Manger‘s 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 1st Monday 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, Dept 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, Dept 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 than the 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.

37. Query to display the department no, name and job for all employees in the Sales department.

CMSDSC510T: Software Engineering Theory 45 Lectures

1. **Introduction** **(06 Lectures)**

The Evolving Role of Software, Software Characteristics, Changing Nature of Software, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella Activities, Process Models, Capability Maturity Model Integration (CMMI).

2. **Requirement Analysis** **(08 Lectures)**

Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS.

3. **Software Project & Risk Management** **(12 Lectures)**

Estimation in Project Planning Process, Project Scheduling Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan.

4. **Quality Management:** **(06 Lectures)**
   Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and Projects.
5. **Design Engineering:** **(07 Lectures)**
   Design Concepts, Architectural Design Elements, Software Architectural Level and Component Level, Mapping of Data Flow into Software Architecture, Modeling Component Level Design.
6. **Testing Strategies & Tactics** **(06 Lectures)**
   Software Testing Fundamentals, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System testing, Black-Box Testing, White-Box Testing and their type, Basis Path Testing.

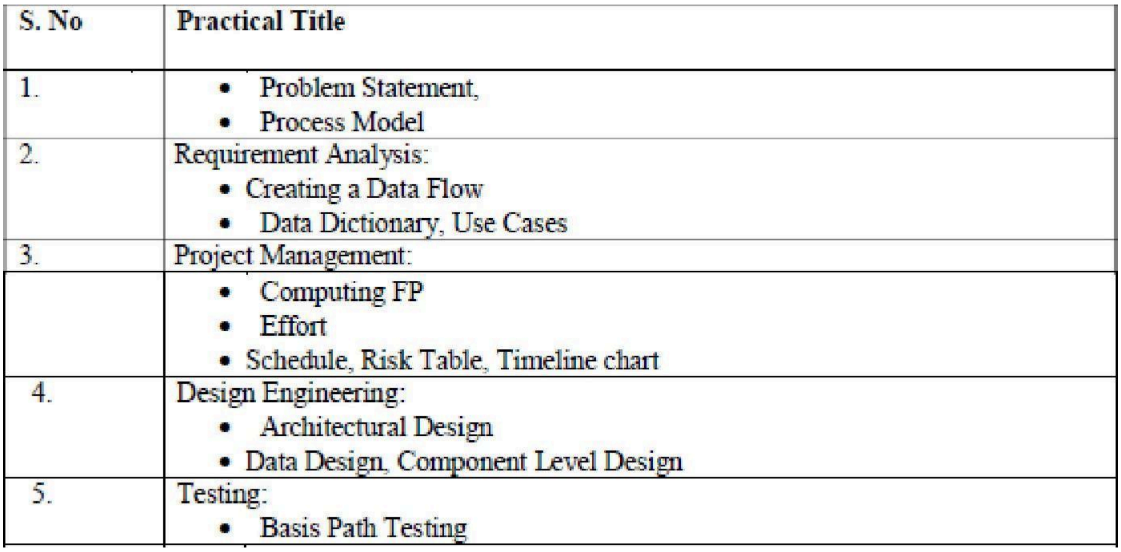
Text Books

1. R.S. Pressman, Software Engineering: A Practitioner‘ s Approach (7th Edition), McGraw-Hill, 2009.
2. P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.

Recommended Books:

1. K.K. Aggarwal and Y. Singh, Software Engineering (2nd Edition), New Age International Publishers, 2008.
2. I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
3. D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
4. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India,2004

CMSDSC510P: Software Engineering **Practical: 60 Lectures**



S. No	Practical Title
1.	• Problem Statement, • Process Model
2.	Requirement Analysis: • Creating a Data Flow • Data Dictionary, Use Cases
3.	Project Management:
	• Computing FP • Effort • Schedule, Risk Table, Timeline chart
4.	Design Engineering: • Architectural Design • Data Design, Component Level Design
5.	Testing: • Basis Path Testing

Sample Projects:

1. **Criminal Record Management**: Implement a criminal record management system for jailers, police officers and CBI officers
2. **DTC 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.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

Projects (Mini Projects) will be assigned to a group of students (maximum 3) under a Faculty member. Students will prepare a presentation and present it in the practical examination.

CMSDSC511T/CMSCOR606T: Networking Theory: 60 Lectures

Additional Tutorial: 15 lectures

1. Introduction to Computer Networks (8 Lectures)

Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

2. Data Communication Fundamentals and Techniques (10 Lectures)

Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation-; multiplexing techniques-FDM, TDM; transmission media.

3. Networks Switching Techniques and Access mechanisms (10 Lectures)

Circuit switching; packet switching- connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

4. Data Link Layer Functions and Protocol (10 Lectures)

Error detection and error correction techniques; data-link control- framing and flow control; error recovery protocols- stop and wait ARQ, go-back-n ARQ; Point to Point Protocol on Internet.

5. Multiple Access Protocol and Networks (5 Lectures)

CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways;

6. Networks Layer Functions and Protocols (6 Lectures)

Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

7. Transport Layer Functions and Protocols (6 Lectures)

Transport services- error and flow control, Connection establishment and release- three way handshake.

8. Overview of Application layer protocol (5 Lectures)

Overview of DNS protocol; overview of WWW &HTTP protocol.

Text Books

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM, 2007.
2. A.S. Tanenbaum: Computer Networks, Fourth edition, PHI, 2002

CMSDSC612T: Graphics and Multimedia Theory: 45 Lectures

1. **Introduction (4 Lectures)**
   Basic elements of Computer graphics, Applications of Computer Graphics.

2. **Graphics Hardware (4 Lectures)**
   Architecture of Raster and Random scan display devices, input/output devices.

3. **Fundamental Techniques in Graphics (15 Lectures)**
   Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

4. **Geometric Modeling (5 Lectures)**
   Representing curves & Surfaces.

5. **Visible Surface determination & Surface rendering (9 Lectures)**
   Hidden surface elimination, Illumination and shading models. Basic color models and Computer Animation.

6. **Multimedia system design & multimedia file handling (8 Lectures)**
   Multimedia basics − Multimedia applications − Multimedia system architecture − Evolving technologies for multimedia − Defining objects for multimedia systems − Multimedia data interface standards − Multimedia databases. Compression and decompression − Data and file format standards − Multimedia I/O technologies − Digital voice and audio − Video image and animation − Full motion video − Storage and retrieval technologies.

Text Books

1. J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice, 2nd edition Publication Addison Wesley 1990.
2. D.Hearn, Baker: Computer Graphics, Prentice Hall of India 2008.

Books Recommended:

1. D.F.Rogers Procedural Elements for Computer Graphics, McGraw Hill 1997.
2. D.F.Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill, 2nd edition 1989.

CMSDSC612P: Graphics and Multimedia Practical: 60 Lectures

Instruction: Use an open source C compiler.

1. Write a program to implement the DDA line drawing algorithm.
2. Write a program to implement Bresenham's line drawing algorithm.
3. Write a program to implement a midpoint circle drawing algorithm.
4. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
5. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
6. Write a program to apply various 2D transformations on a 2D object (use homogeneous coordinates).
7. Write a program to apply various 3D transformations on a 3D object and then apply parallel and

perspective projection on it.

8. Write a program to draw Hermite/Bezier curve.

CMSDSC613T: Artificial Intelligence Theory: 45 Lectures

1. **Introduction** **(06 Lectures)**
   Introduction to artificial intelligence, background and applications, Turing test and rational agent approaches to AI, agents & environment, nature of environment, structure of agents, goal-based agents, intelligent agent, learning agents.
2. **Problem Solving by Searching** **(08 Lectures)**
   Problems, Defining the problem as state space search, production system, problem characteristics, issues in the design of search programs, solving problems by searching, problem solving agents. searching for solutions; Uninform search strategies: Breadth first search, Depth first search, Depth limited search.
3. **Heuristic Search Strategies** **(06 Lectures)**
   Greedy best first search, A* algorithm. Memory bounded heuristic search: local search algorithms & optimization problems: Hill climbing and its variations, constraint satisfaction problem, Means-End analysis.
4. **Adversarial Search:** **(05 Lectures)**
   Introduction to game playing, optimal decisions and strategies in games, Min-Max and Alpha-Beta pruning algorithms.
5. **Knowledge Representation** **(20 Lectures)**
   Introduction to First Order predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, conceptual graphs. Programming in Logic (PROLOG)

Text Books

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell &Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd edition, 2005.

Recommended Books:

1. Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2nd edition, 1991.
2. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.
3. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd edition, 2000.

CMSDSC613P: Artificial Intelligence Lab: 60 Lectures

Instruction: Use an open source Prolog compiler.

1. Write a prolog program to calculate the sum of two numbers.
2. Write a prolog program to find the maximum of two numbers.
3. Write a prolog program to calculate the factorial of a given number.
4. Write a prolog program to calculate the nth Fibonacci number.
5. Write a prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n‘th element into every list at all levels.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program, remove-nth(Before, After) that asserts the After list is the Before list with the removal of every n‘th item from every list at all levels.
8. Write a Prolog program to implement append for two lists.

9. Write a Prolog program to implement palindrome (List).
10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.
15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.
16. Write a Prolog program to implement GCD of two numbers.

CMSDSC614T: Microprocessor and VLSI Design Theory: 45 Lecture

1. Microprocessor architecture (8085): 12 Lectures

Internal architecture, system bus architecture, memory and I/O interface.

Evolution of Microprocessor its advantages and disadvantages, Basic Architecture of Microprocessor 8085 and explanation of each block, Register sets, Microprocessor 8085 pin out and signals, Instruction Formats, Instruction Cycle, address lines calculations, Clock Cycle, Time period calculation of each State, Multiplexed Address Data Bus, Control and Status signals, Microprocessor and Bus Timing, De-multiplexing of Address Data Bus, Generation of Control Signals for I/O and Memory, Basic concepts in Memory Interfacing, Address Decoding and memory Addresses.

2. Microprocessor programming (8085): 8 Lectures

Register Organization, instruction formats, assembly language programming.

Instruction Set of 8085, Different Programming Techniques, Stack and Subroutines, Counter and Time Delays, Code Conversion, BCD Arithmetic and 16 bit Data Operation.

3. Interfacing (8085): 8 Lectures

Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces.

Basic Interfacing concepts, Peripheral I/O instructions (I/O mapped I/O), Device Selection and data Transfer, Absolute and Partial Decoding, Input Interfacing, Interfacing I/O using decoders, Memory mapped I/O techniques, Data transfer schemes, Interfacing 8155 memory segment.

4. Interrupts (8085): 4 Lectures

Interrupt, RST instructions, Software and Hardware interrupt, multiple Interrupts and Priorities, 8085 Vectored Interrupts, Restart as Software Instructions. Interfacing Digital to Analog Converters, Analog to Digital Interfacing, keyboard interfacing, interfacing 8255 (Mode - 0, BSR), Support IC chips- 8237/8257,8259.

5. Microprocessor 8086: 3 Lectures

The 8086 microprocessor- Basic Architecture, Instruction set, Addressing modes.

6. Introduction to VLSI Design: 10 Lectures

Introduction: Switching circuits, gradual development of transistor technology,

SSI, MSI, LSI, VLSI.

MOSFET: Basic science, as switch, synthesis rule: n- complex, p- complex, CMOS Structure, switching circuit design using nmos, cmos , stick diagram

Introduction to Floor Planning

Text Books

1. Microprocessor architecture, programming and applications with 8085/8085A, Ramesh Gaonkar, Penram International Publication (PRI).
2. Fundamental of Microprocessors and Microcontrollers, B.Ram, Dhanpat Rai Publications.

Books Recommended:

1. Microprocessors and Microcontrollers, Senthil, Saravanan, Jeevananthan, Oxford.
2. Advanced Microprocessors and Peripherals by Ray and Bhurchandi – McGrawHill.
3. Intel Corp. Micro Controller Handbook – Intel Publications.
4. Microprocessors and Interfacing Programming and hardware by Douglas V. Hall, McGraw Hill.
5. Microprocessor 8085 and its Interfacing, Mathur, PHI..
6. The Intel Microprocessors: 8086/8088, 80186, 80286, 80386 & 80486, Bary B. Brey, Prentice Hall, India.

CMSDSC614P: Microprocessor and VLSI Design 60 Lecture

1. Assembly Language Programming for Arithmetic Operations like Addition, Subtraction, Multiplication and Division on 8, 16 bit data.

2. Assembly Language Programming for different logical operations.

3. Assembly Language Programming for code conversions.

4. Assembly Language Programming for different sorting techniques.

5. Assembly Language Programming for memory block transfer.

6. Assembly Language Programming for AP series and Fibonacci series.

7. Assembly Language Programming for Searching.

8. Assembly Language Programming for frequency distribution.

9. Block Replacement and transfer

Many more programs can be included related to the programming techniques of Microprocessor 8085 of similar type and level

CMSDSC615T: Theory of Computation Theory: 60 Lectures

Additional Tutorial: 15 Lectures

1. **Languages** **(8 Lectures)**

   Alphabets, string, language, Basic Operations on language, Concatenation, Kleene Star

2. **Finite Automata and Regular Languages** **(20 Lectures)**

   Deterministic (DFA) and Non-deterministic finite automata (NDFA), NFA to DFA Conversion, Regular languages and their relationship with finite automata. Regular Expressions and their relationship with finite automata. Pumping lemma and closure properties of regular languages.

3. **Context free languages** **(16 Lectures)**

   Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Non-deterministic), Acceptance, Relationship of it with CFG and conversions. Properties of context free languages (concept only) and Normal forms.

4. **Turing Machines and Models of Computations** **(16 Lectures)**

Turing Machine as a model of computation, Definition, Instantaneous Description (ID), Moves between ID's. Acceptance of a language by a TM. Universal Turing Machine, Language acceptability, decidability, halting problem, Recursively enumerable and recursive languages

Text Books

1. Daniel I.A.Cohen, Introduction to computer theory, John Wiley,1996
2. Lewis & Papadimitriou, Elements of the theory of computation, PHI 1997.

Recommended Books:

1. Hopcroft, Aho, Ullman, Introduction to Automata theory, Language & Computation – Third Edition, Pearson Education. 2006
2. P. Linz, An Introduction to Formal Language and Automata fourth edition Publication Jones Bartlett, 2006

CMSDSC716T: Image Processing and Computer Vision Theory: 45 Lectures

1. Introduction (4 Lectures)

Light, Brightness adaption and discrimination, Pixels, Coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, Sampling and quantization.

2. Spatial Domain Filtering: (8 Lectures)

Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, Sharpening filters, Gradient and Laplacian.

3. Filtering in the Frequency domain: (8 Lectures)

Hotelling Transform, Fourier Transforms and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering.

4. Image Restoration: (8 Lectures)

Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections.

5. Image Compression & Segmentation: (10 Lectures)

Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Run length coding. JPEG. Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, Local processing, Regional processing, Hough transform, Thresholding, Iterative thresholding.

6. Image Description (5 Lectures)

Introduction to Computer Vision: Comparison of Image Processing, Computer Vision and Computer Graphics, What is Computer Vision - Low-level, Mid-level, High-level processing, Overview of Diverse Computer Vision Applications: Document Image Analysis, Biometrics, Object Recognition, Object Tracking, Gesture Recognition, Motion Estimation

Text Books

1. R C Gonzalez , R E Woods, Digital Image Processing, 4th Edition, Pearson Education.2018.

2. M A Joshi, Digital Image Processing: an algorithmic approach, 2nd Edition, PHI,

Reference Books:

1. K R Castleman, Digital Image Processing, Pearson Education.1996

2. Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons. 1989.

CMSDSC716T: Image Processing and Computer Vision Practical: 60 Lectures

Instruction: Use an open source Python compiler with OpenCV.

1. Write a program to read and display digital image.

a. Become familiar with Basic commands

b. Read and display image in

c. Resize given image

d. Convert given color image into gray-scale image

e. Convert given color/gray-scale image into black & white image

f. Draw image profile

g. Separate color image in three R G & B planes

h. Create color image using R, G and B three separate planes

i. Flow control and LOOP

j. Write given 2-D data in image file

2. To write and execute image processing programs using point processing method

a. Obtain Negative image

b. Obtain Flip image

c. Obtain B/W image by Thresholding

d. Contrast stretching

3. To write and execute programs for image arithmetic operations

a. Addition of two images

b. Subtract one image from other image

c. Calculate mean value of image

d. Different Brightness by changing mean value

4. To write and execute programs for image logical operations

a. AND operation between two images

b. OR operation between two images

c. Calculate intersection of two images

d. Watermarking using EX-OR operation

e. NOT operation (Negative image)

5. To write a program for histogram calculation and equalization using

a. Standard function

b. Program without using standard functions

6. To write and execute program for geometric transformation of image

a. Translation

b. Scaling

c. Rotation

d. Shrinking

e. Zooming

7. To understand various image noise models and to write programs for

a. image restoration

b. Remove Salt and Pepper Noise

c. Minimize Gaussian noise

d. Median filter and Weiner filter

8. Write and execute programs to remove noise using spatial filters

a. Understand 1-D and 2-D convolution process

b. Use 3x3 Mask for low pass filter and high pass filter

9. Write and execute programs for image frequency domain filtering

a. Apply FFT on given image

b. Perform low pass and high pass filtering in frequency domain

c. Apply IFFT to reconstruct image

10. Write a program for edge detection using different edge detection mask

11. Write and execute a program for image morphological operations erosion and dilation.

12. To write and execute a program for wavelet transform on a given image and perform inverse wavelet transform to reconstruct the image.

CMSDSC717T: Compiler Design Theory: 60 Lectures

Additional Tutorial: 15 lectures

1. Introduction to Compilers : (12 Lectures)

Introduction, Translators, Compilation and Interpretation, Language processors, The Phases of Compiler : Lexical Analysis, Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens , Lex, Finite Automata, Regular Expressions to Automata, Minimizing DFA.

2. Syntax Analysis: (12 Lectures)

ambiguity, associativity, precedence, Role of Parser, Top down Parsing, recursive-descent parsing, predictive parsing, Bottom up Parsing, LL(1) Parser-Shift Reduce Parser, Operator precedence grammar, LR parsers Syntax directed definitions: inherited and synthesized attributes.

3. Intermediate Code Generation: (12 Lectures)

Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking.

4. Run-Time Environment and Code Generation: (12 Lectures)

Storage Organization, Stack Allocation Space, Access to Non-local Data on the Stack, Heap Management, Issues in Code Generation, Symbol Tables. Runtime systems, Activation tree, Activation record, Basic Blocks, Dataflow analysis,

5. Code Optimization: (12 Lectures)

Principal Sources of Optimization, Peep-hole optimization, DAG- Optimization of Basic Blocks-Global Data Flow Analysis, Efficient Data Flow Algorithm.

Text Book:

1. Aho, Compilers: Principals, Techniques and Tools, Pearson Education.
2. Muneeswaran, Compiler Design,Oxford

References:

1. Sudha Sadasivam, Compiler Design - 2nd Edn., SCITECH
2. Niklaus Wirth, Theory and Techniques of Compiler Construction, Addison-Wesley

CMSDSC818T: Machine Learning Theory: 45 Lectures

1. Introduction to Machine Learning: (8 Lectures)

Motivation and role of machine learning in computer science and problem solving, Data and Features representation, Feature reduction/Dimensionality reduction and PCA (Eigen values, Eigen vectors, Orthogonality), Introduction to paradigms of Learning (supervised, unsupervised, and reinforcement learning). Cross Validation, Two-Class Evaluation Measures, The ROC Curve, Minimum Description Length & Exploratory Analysis.

2. Supervised Learning: (12 Lectures)

Notion of Training, Validation and Testing; Connect to generalization and overfitting. Linear Regression, Multivariate Regression, Logistic Regression. Decision trees, Naive Bayes, Bayesian Networks, Support Vector Machine (SVM), Nearest Neighbours and KNN. Introduction Artificial Neural Network, Early model, Role of Loss Functions and Optimization, Gradient Descent and Perceptron/Delta Learning, Multilayer Perceptron (MLP), Backpropagation.

3. Unsupervised Learning: (8 Lectures)

Partitional Clustering- K-means, Hierarchical Clustering- Agglomerative and Divisive, Density Based Clustering - DBSCAN, Expectation Maximization - Frequent itemset generation, Association rule mining (Apriori Algorithm).

4. Graphical Models: (6 Lectures)

Potential Functions, Hidden Markov Models, Variable Elimination.

5. Combining Multiple Learners: (6 Lectures)

Generating diverse learners, model combination schemes, voting, error-correcting output codes, Random forest, bagging, boosting, bootstrapping, mixture of experts revisited, stacked generalization, fine-tuning an ensemble, cascading

6. Introduction to Deep Learning: (5 Lectures)

Convolution neural network (CNN), Layers in CNN, CNN architectures, brief introduction to other deep learning models.

Text Books:

1. Introduction to Machine learning, Ethem Alpaydin, Third Edition, MIT Press, 2009.

2. Machine learning for dummies, John Paul Muller, Luca Massaron, Weily, 2 nd Edition, 2021.

Reference:

1. Introduction to Machine Learning with Python A guide for data scientists, Andreas, C. Muller Sarah Guido, O'Reilly

CMSDSC818P: Machine Learning Practical: 60 Lectures

List of Exercises for Machine Learning Lab:

Instruction: Use an open source Python compiler

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV le

2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm. Output a description of the set of all hypotheses consistent with the training examples.

3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample

4. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets

5. Write a 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. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library lasses/API.

8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using the k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

9. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set.

10. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

11. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points.

12. Select appropriate data set for your experiment and draw graphs

CMSDSC819T: Cyber Security and IoT **Theory: 60 Lectures**

Additional Tutorial: 15 lectures

1. Introduction to Cyber Security (8 Lectures)

Cyber Security definition and Concepts, OSI Security Architecture, Security Objectives and Challenges, Security Attacks- active and Passive attack, Security Services, Security Mechanisms, Fundamental Security Design Principles, Cyber Security model.

2. Cryptography and Digital signature (12 Lectures)

Classical Encryption Techniques-Symmetric Cipher Model, Substitution Techniques, Transposition Techniques and Steganography. Public-Key Cryptography and RSA algorithm, Digital Signatures Requirements, properties and mechanism.

3. Malicious Software (8 Lectures)

Introduction to Malicious Software (Malware), Types of Malware- Viruses, Worms, Spam E-mail, Trojans, Keyloggers, Phishing and Spyware. Propagation of Viruses, Worms and Trojans. Countermeasures, Denial of Service Attacks (DOS).

4. Firewalls and Intrusion Detection System (12 Lectures)

Intruders, Intrusion Detection, Password Management. Need for Firewalls, Firewall Characteristics and Access Policy, Types of Firewalls, Firewall Basing, Firewall Location and Configurations.

5. Legal and Ethical Aspects (8 Lectures)

Cybercrime and Computer Crime, Intellectual Property, Privacy and Ethical Issues

6. Internet of Things (IoT) (12 Lectures)

An overview and evolution of Internet of Things, Components of IoT-Enabled Things, IoT Security Concepts and Objectives, Tamper Resistance and Detection Gateway Security, IoT Security Environment, An Open-Source IoT Security Module- Cryptographic Algorithms
Operating Modes and Offset Codebook Mode.

Text Books

1. Cryptography and Network Security-Principles and Practice, William Stallings, Pearson/Prentice Hall
2. Introduction to Cryptography and Network Security, Behrouz A. Forouzan, McGraw Hill.

Reference Books

1. Introduction to Cyber Security Guide to the World of Cyber Security, Anand Shinde, Notion Press.

CMSDSC820T : Big Data **Theory: 60 lectures**

Additional Tutorial: 15 lectures

1. UNDERSTANDING BIG DATA (10 lectures)

What is big data – why big data –.Data!, Data Storage and Analysis, Comparison with Other Systems, Relational Database Management System, Grid Computing, Volunteer Computing, convergence of key trends – unstructured data – industry examples of big data – web analytics –big data and marketing – fraud and big data – risk and big data – credit risk management – big data and algorithmic trading – big data and healthcare – big data in medicine – advertising and big data – big data technologies – introduction to Hadoop – open source technologies – cloud and big data – mobile business intelligence – Crowd sourcing analytics – inter and trans firewall analytics.

2. NOSQL DATA MANAGEMENT (20 lectures)

Introduction to NoSQL – aggregate data models – aggregates – key-value and document data models – relationships – graph databases – schema less databases – materialized views –distribution models – shading –– version – map reduce – partitioning and combining –composing map-reduce calculations.

3. BASICS OF HADOOP (18 lectures)

Data format – analyzing data with Hadoop – scaling out – Hadoop streaming – Hadoop pipes –design of Hadoop distributed file system (HDFS) – HDFS concepts – Java interface – data flow - Hadoop I/O – data integrity – compression – serialization – Avro – file-based data structures.

4. MAPREDUCE APPLICATIONS (15 lectures)

MapReduce workflows – unit tests with MRUnit – test data and local tests – anatomy of MapReduce job run – classic Map-reduce – YARN – failures in classic Map-reduce and YARN - job scheduling – shuffle and sort – task execution – MapReduce types – input formats – output formats

5. HADOOP RELATED TOOLS (12 lectures)

Hbase – data model and implementations – Hbase clients – Hbase examples –praxis. Cassandra Cassandra data model – Cassandra examples – Cassandra clients –Hadoop integration. Pig –Grunt – pig data model – Pig Latin – developing and testing Pig Latin scripts. Hive – data types and file formats – HiveQL data definition – HiveQL data manipulation – HiveQL queries.

Text Books

1. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
2. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.

Reference Books:

1. Vignesh Prajapati, Big data analytics with R and Hadoop, SPD 2013.
2. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
3. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.
4. Alan Gates, "Programming Pig", O'Reilley, 2011

MDC IN COMPUTER Science

CMSHMD101T/CMSGMD101T CMSHMD201T/CMSGMD201T CMSHMD301T/CMSGMD301T (Credits:3) : Computer Fundamentals 45 Classes

Introduction to Computer: 11 classes

Computer Characteristics, Concept of Hardware (Functional Block Diagram of Computer. CPU, Input Device like Keyboard, Mouse, Scanner, MICR, OMR, Light Pen, Joysticks etc. Output Device: VDU, Printers – Dot Matrix, Daisy-wheel, Inkjet, Laser, Line Printers and Plotters. Computer Memory: Primary Memory, Secondary Memory, Cache Memory) Software (System and Application Software), Evolution of computer and Generations, Types of Computer – Analog and Digital computers, Hybrid Computers, General Purpose and Special Purpose Computer, Limitations of Computer Applications of Computer in Various Fields.

Operating System: **11 classes**

Operating System, Evolution of Operating System. Functions of Operating Systems. Batch Processing, Time Sharing, Multi-programming, Multi-Processing. Types of Operating Systems. Detailed Study of Windows Operating System (Basic components of windows, icons, types of icons, taskbar, activating windows, using desktop, title bar, running applications, exploring computer, managing files and folders, copying and moving files and folders. Control panel—display properties, adding and removing software and hardware, setting date and time, screen saver and appearance. Using window accessories.). Introduction and Features of LINUX OS.

Networking: **12Classes**

Concept, Basic Elements of a Communication System, Data Transmission Media, Data Transmission Modes(Simplex, Half-duplex, Full-duplex), Network Topologies, Network Types: LAN, MAN, WAN Intranet and Internet, Modem, Bridges, Routers, Concept of Channel, Bandwidth (Hz, KHz, MHz) and Data Transfer rate (bps, Kbps, Mbps, Gbps, Tbps), Internet applications. World Wide Web, E-mail (E-mail Message Components, Advantages, Disadvantages), browsing and searching, search engines, Understanding URL, Client and Server Computers

Computer Security: **5 Classes**

An Overview of Computer Security, Computer virus (Definition, Characteristics, types, Impact), Virus Preventive Measures: antivirus and Firewall .

Brief understanding of the following emerging trends: (Basic Concept) **6 Classes**

Artificial Intelligence, Robotics, Internet of Things (IoT), Sensors, Smart cities, Cloud Computing and Cloud Services (SaaS, IaaS, PaaS); Blockchain technology.

Text Books:

1. Computer Fundamentals by Priti Sinha, Pradeep K., Sinha.
2. Fundamentals of Computers by Rajaraman V and Adabala N.
3. Fundamentals of Computers by Reema Thareja

SEC IN COMPUTER APPLICATION

CMSHSE101M/CMSGSE101M: **(Credits:3):**
R Programming **45 Classes**

Module 1: Introduction to R Programming 6 Classes

1. Introduction to R
   - Overview of R and its uses in data analysis and statistics.
   - Installing R and RStudio.
   - Introduction to RStudio interface and its components (Console, Source, Environment, Plots,

etc.).
- Understanding the R command prompt.
- Basic operations: arithmetic operations, logical operations, and comparison operators.
- Data types in R: numeric, integer, character, logical, and factor.
- Variable assignment and naming conventions.

2. Data Structures in R
   - Vectors: creation, indexing, and basic functions (length, class, typeof, etc.).
   - Lists: creation, indexing, and subsetting.
   - Matrices: creation, indexing, and operations.
   - Data frames: creation, importing/exporting data, indexing, and subsetting.
   - Factors: creation, levels, and usage in categorical data.

Module 2: Data Manipulation and Management 10 Classes

1. Data Import and Export
   - Reading data from CSV, Excel.
   - Writing data to CSV, Excel.
2. Data Cleaning and Preparation
   - ○ Handling missing values and duplicates.
   - ○ Data type conversions.
   - ○ Renaming columns and rows
3. Data Transformation
   - ○ Selecting columns (select), filtering rows (filter), arranging data (arrange).
   - ○ Mutating and transforming data (mutate, transmute).
   - ○ Summarizing data (summarize, group_by).

Module 3: Data Visualization 10 Classes

1. Introduction to Data Visualization in R
   - ○ Understanding the basics of data visualization.
   - ○ Using base R graphics for plotting.

2. Using ggplot2 for Advanced Visualization
   - ○ Introduction to the grammar of graphics.
   - ○ Creating basic plots (scatter plots, line plots, bar plots, histograms).
   - ○ Customizing plots (titles, labels, themes, colors, and scales).
   - ○ Faceting and multi-plot layouts.

3. Interactive Visualizations
   - ○ Introduction to interactive plots using plotly and shiny.
   - ○ Creating basic interactive plots and dashboards.

Module 4: Statistical Analysis and Modeling 3 Classes

- Descriptive Statistics
  - ○ Calculating measures of central tendency (mean, median, mode).
  - ○ Calculating measures of dispersion (range, variance, standard deviation).

Module 5: Advanced R Programming 10 Classes

1. Control Structures
   - ○ Conditional statements (if, else, switch).

- Looping constructs (for, while, repeat).
- Vectorized operations and the apply family of functions (apply, lapply, sapply, tapply, mapply).

2. Debugging and Error Handling

   - Debugging tools in R (debug, trace, browser).
   - Handling errors and warnings (try, tryCatch).

Simple Programs : 6 Classes

1. Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.
2. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.
3. Write a R program to multiply two vectors of integers type and length 3.
4. Write a R program to find Sum, Mean and Product of a Vector, ignore elements like NA or NaN.
5. Write a R program to list containing a vector, a matrix and a list and give names to the elements in the list.
6. Write an R program to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame.
7. Write a R program to sort a given data frame by multiple column(s).
8. Write a R program to compare two data frames to find the row(s) in the first data frame that are not present in the second data frame.
9. Write a program to read a csv file and find min, max and range the data in the file in R
10. Write a program to find Mean, Median and Mode

Text Books:

1.William N. Venables and David M. Smith, An Introduction to R. 2nd Edition. Network Theory Limited.2009

2. Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

Reference Books:

1.The Book of R,Tilman M. Davies,No Starch Press,1st edition

2.Discovering Statistics Using R,Andy Field,SAGE Publications Ltd,1st edition

CMSHSE202M/CMSGSE202M: **(Credits:3)**

Python Programming **45 Classes**

1. Planning the Computer Program: **5 Classes**

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.

2. Techniques of Problem Solving **5 Classes**

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

3. Overview of Programming **6 Classes**

Structure of a Python Program, Elements of Python

4. Introduction to Python **4 Classes**

Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator).

5. **Creating Python Programs** **10 Classes**

Input and Output Statements, Control statements(Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments.

6. **Simple programs** **15 Classes**

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.

2. WAP 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 >=80
Grade B:
Percentage>=70
and <80
Grade C:
Percentage>=60
and
<70
Grade D:
Percentage>=40
and <60
Grade E:
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 the user.
4. WAP to display the first n terms of Fibonacci series.

5. WAP to find the factorial of the given number.

6. WAP to check whether a given number is prime or not.
7. WAP to check whether a given string is palindrome or not.
8. WAP to find the sum of the following series for n terms: 1 + 1/2 + 1/3 …

Text Books :

1.John V. Guttag, “Introduction to Computation and Programming Using Python”, MIT Press
2. Allen Downey, “Think Python: How to Think Like a Computer Scientist”, O’Reilly
3. Mark Lutz, “Learning Python, 5th Edition”, O’Reilly

Reference Books:

1.. Python Programming for the Absolute Beginner, Michael Dawson, Cengage Learning.
2.. Learning to Program in Python 2017, P. M. Heathcote, PG Online Limited

3. Python Programming Fundamentals: Lee and Kent D.