PART – A

1. **Complex Variables**: Functions of complex variables, Limit, continuity and differentiability (definitions only), Analytic functions, Cauchy-Reimann equations in Cartesian and Polar forms–Consequences, construction of analytic function (Cartesian and polar forms), Definition of Conformal transformations: \( z^2, e^z \) and \( z + a^2 / z \ (z \neq 0) \), Bilinear transformations.

2. **Complex Integrations**: Line integral, Cauchy’s theorem–corollaries, Cauchy’s integral formula. Taylor’s and Laurent’s series (statements only), Singularities, Poles, calculation of Residues, Residue theorem (without proof)–problems.

   **14 Hours**.

PART – B

1. **Special Functions**: Series solution of Bessel’s differential equation, recurrence formulae, Generating function, orthogonal property, Bessel’s integral formula. Series solution of Legendre’s differential equation, Recurrence formulae, Generating function, orthogonal property, Rodrigue’s formula

   **14 Hours**.
PART – C

1. **Statistics & Probability**: Curve fitting by the method of least squares: \( y = a + bx, \ y = abx, \ y = a + bx + cx^2 \). Correlation and Regression. Probability- addition rule, conditional probability Multiplication rule, Bayes” rule. Discrete and continuous random variables-PDF-CDF, Binomial, Poisson, Exponential and Normal distributions.

12 Hours.

PART – D

1. **Sampling distribution**: Sampling, Sampling distribution, Standard error, Null and alternate hypotheses, Type I and Type II errors, Testing of hypothesis for Means, Level of Significance for Means, Confidence limits for Means, large and small samples, Student’s t-distribution.


14 Hours.


PART – A : Complex Variables:
Chapter 20 : 20.1 to 20.6, 20.8 (4), 20.9 (1), 20.10 (1 to 3) 20.12, to 20.14, 20.16 (2,3) 20.17 to 20.19

PART – B : Special Functions:
Chapter 16 : 16.6 to 16.9, 16.11, 16.13, to 16.17

PART – C : Statistics and Probability:
Chapter 1:1.13,1.14
Chapter 23:23.9,23.10,23.14,23.16 to 23.21
PART – D: Sampling Distributions:
Chapter 23:23.31 to 23.33, 23.34, to 23.36
Joint Probability distributions and Markov Chains:


---

GRAPH THEORY & COMBINATORICS
(CSE / ISE)

Sub code: CSE 42
Exam Marks: 100
Hours/week: 4
IA Marks: 25
Total Hours: 52
Exam Hours: 3

PART – A

1. Introduction to Graph Theory: Directed and Un-directed graphs-
Basic Terminologies, Walks and Paths, Connected and
Disconnected graphs. Subgraphs., Complete graph, Compliment
of a graph and Graphs isomorphism. Vertex Degree: Degree of a
vertex, Regular graph, Konigsberg’s bridge problem, Eularian
and Hamiltonian graphs. Traveling salesman problem.
7 hours

2. Planar Graphs: Planar graphs-Definition and examples. Bipartite
and complete Bipartite graphs, Kuratowski’s graphs, Euler’s
formula, Detection of planarity of a graph. Dual of a planar
graph, Observations about a planar graph and its dual. Graph
Coloring-Property Coloring of a Graph, Chromatic Number,
Chromatic Polynomial, Four color problem – only discussion.
06 Hours.

3. Trees: Definition of a tree, Properties of a tree. Rooted trees and
Binary rooted trees, examples. Trees and sorting, Weighted Trees
and Prefix Codes. Spanning trees.
06 Hours.

07 Hours

PART – B

COMBINATORICS


07 Hours.


07 Hours.


07 Hours.


06 Hours.

TextBooks:

Part – A : Graph Theory :
Chapter : 11.1 to 11.6 , 12.1 to 12.4 , 13.1 to 13.4.
Part – B : Combinatorics :
Chapter : 1.1 to 1.6 , 5.3 , 5.5 , 8.1 , to 8.5 , 9.1 , to 9.5 , 10.1 to 10.4

2 ) Theory and Problems of Combinatorics including concept of Graph Theory by V.K.Balakrishnan (Schaum’s outline series), Mcgraw Hill ,1995

Part – B : Combinatorics :
Chapter : 1.1 to 1.3 , Solved Problems , 2.1 to 2.3 Solved Problem and 3.1 to 3.5 and Solved Problems .

Reference Books : Graph Theory with Applications to Engineering and Computer Science by Narsingh Deo, PHI Publications (1986).

Note : The Question paper consists of two parts A & B Each part will contain four Questions . Part – A will be on Graph Theory and Part – B will be on Combinatorics . The student are expected to answer a total of five full Questions choosing at least two full questions from each part. .from each part .

ANALYSIS AND DESIGN OF ALGORITHMS
(CSE and ISE)

Sub code : CSE43
Hours/week: 4
Total Hours: 52

Exam Marks: 100
IA Marks: 25
Exam Hours: 3

PART – A

   06 HOURS .

2. Fundamental of the Analysis of Algorithm Efficiency : Analysis Frame work , Asymptotic Notation and Basic efficiency classes ,
Mathematical analysis of Recursive and Nonrecursive algorithms
Examples .

06 HOURS .

3. Brute Force : Selection, Sort and Bubble Sort, Sequential Search
and String Matching, Exhaustive Search.

03 HOURS .

4. Divide And Conquer : Merge sort, Quick sort, Binary search,
Binary tree Traversals and related properties, Multiplication of
large integers, Stressen’s Matrix Multiplication.

05 HOURS .

5. Decrease and Conquer : Insertion sort, Depth First and Breadth
First Search, Topological Sorting, Algorithms for Generating
Combinatorial Objects.

05 HOURS .

6. Transform and Conquer : Presorting, Balanced Search Trees,
Heaps and Heapsort, Problem Reduction.

05 HOURS .

7. Space and Time Tradeoffs : Sorting by Counting, Input
Enhancement in string Matching, Hashing.

04 HOURS.

8. Dynamic Programming : Computing a binomial coefficient,
Warshall’s and Floyd’s Algorithms, The knapsack problem and
memory functions.

04 HOURS.

9. Greedy Technique : Prim’s Algorithm, Kruskal’s Algorithm,
Dijkstra’s Algorithm, Huffman Trees.

05 HOURS.

10. Limitations of Algorithm Power : Lower-bound Arguments,
Decision Trees, P, NP and NP-Complete Problems .

04 HOURS.

05 HOURS.


Chapter: 1.1 – 1.4, 2.1 – 2.5, 3.1, 3.2, 3.4, 4.1 – 4.5, 5.1 – 5.4, 6.1, 6.3, 6.4, 6.6, 7.1 – 7.3, 8.1, 8.2, 8.4, 9.1 – 9.4, 10.1 – 10.3, 11.1 – 11.3.

Reference Books:

PART – A

1. Introduction To Finite Automata: Introduction to Finite Automata, the central concepts of Automata theory, deterministic finite automata, Non deterministic Finite automata, an application, Finite automata with Epsilon-transitions.

9 HOURS.

2. Regular Expression and Languages, Properties Of Regular Languages: Regular expressions, Finite Automata and Regular expression, Applications of Regular Expressions, Proving languages not to be regular, Closure properties of regular
languages. Decision properties of Regular languages, Equivalence and minimization of automata.

9 HOURS.


9 HOURS.

4. Pushdown Automata: Definition of the pushdown Automata, The languages of a PDA, Equivalence of PDA’s and CFG’s, Deterministic Pushdown Automata.

8 HOURS.

5. Properties Of Context – Free Languages: Normal forms for CFG’s, The pumping Lemmas for CFG’s, Closure properties of CFL’s.

4 HOURS.

6. Introduction to Turing Machines: Problems that computers cannot solve, The Turing Machine, Programming techniques for Turning Machines, Extensions to the basic Turing Machines, Restricted Turing Machines, Turing Machine and Computers.

9 HOURS.

7. Undecidability: A Language that is not recursively enumerable, An Undecidable problem that is RE, Post’s Correspondence problem, Other undecidable problems.

4 HOURS.


Chapter: 1.1, 1.5, 2.1 – 2.5, 3.1 – 3.3, 4.1 – 4.4, 5, 6, 7.1 – 7.3, 8, 9.1, 9.2, 9.4, 9.5, expect 6.4.4.
MICROPROCESSORS
(CSE, ISE, EE, EC, IT, TC, BM and ML)

Sub code: EC45
Exam Marks: 100
Hours/week: 4
IA Marks: 25
Total Hours: 52
Exam Hours: 3

PART – A

1. Architecture and Operations: Introduction to 8085,
Microprocessor organization/architecture & its operation,
Microprocessor-based system, Memory interfacing, basic
interfacing concepts, Interfacing I/O devices.

05 HOURS.

2. Programming the 8085: Programming model, Instruction
classification, Instruction Format, addressing modes, Writing
Assembly level programs - overview of instruction set, timing
diagrams. Data Transfer, Arithmetic, Logic branch operations.
Programming Techniques-Looping, Counting and Indexing, 16 bit
arithmetic operations, logic operations, Compare and rotate
operations. Stacks and subroutines - Conditional CALL and
RETURN instructions, Advanced subroutine concepts, BCD to
Binary and Binary to BCD conversions, BCD to 7 Segment
conversion, Binary to ASCII and ASCII code conversion, BCD
addition and subtraction, multiplication and division.

21 HOURS.

3. Memory Interface: Memory and I/O mapping and interfacing
concepts.

05 HOURS.

4. Interrupts: 8085 vectored interrupts, Restart as software
instruction, Additional I/O concepts and processors.

06 HOURS.
5. Interfaceing of Peripherals (I/Os) and applications: Interfacing Keyboard (linear and matrix) and 7 segment display including multiplexes, The 8297 Programmable keyboard / display interface. 8255 Programmable Peripheral Interface, 8253 interfacing, 8259 Programmable interrupt controller, DMA and 8257 DMA controller, Serial Communication using 8251, D to A converters, A to D converters and interfacing. RS232 serial Communication standards.

15 HOURS.

Text Books:

Reference Books:

Convergence in the text book:
Architecture and Operations: Ch1 – 1.1-3 : Ch2 – 2.1 – 3 : Ch3 – 3.1
Programming the 8085: Ch3 , 3.21 – 25 : Ch5 , 5.1 – 5 : Ch6 , 6.1 – 6;
Memory Interface: Ch 3.3 ; Ch 4.1 – 4 ;
Interrupts : Ch 12 – 12.1 – 4
Interfacing of Peripherals (I/O's) and applications: Ch 14 – 14.1 , 14.3 ; Ch 15 – 15.1 , 15.2 , 15.4 – 6 ; Ch 16.4
1. **Basic Structure Of Computers**: Computer types: Functional Units, Input Unit, Memory Unit, Arithmetic & logic unit, Output unit, Control unit; Basic Operational Concepts, Bus Structures, Performance, Processor clock, Basic Performance equation, Pipelining & superscalar operation, Clock rate, Performance measurement: Multiprocessor & Multicomputers: Historical Perspective: The First generation, The second generation, the Third generation, the Fourth generation, Beyond the Fourth generation, Evaluation of Performance.

   **04 HOURS**.


4. **The Memory System**: Some Basic Concepts, Semiconductor RAM Memories: Internal Organization of Memory chips, Static Memories, Asynchronous DRAMs, Synchronous DRAMs, structure of Larger Memories, Memory System Consideration, Rambus memory; Read-Only Memories, ROM, PROM, EPROM, EEPROM, Flash memory: Speed, Size & Cost: Cache Memories: Mapping functions; Performances considerations; Interleaving, Hit Rate & Miss Penalty; Virtual memories: Address Translation: Secondary Storage: Magnetic Hard Disks, Optical Disks.


6. **Basic Processing Unit**: Some Fundamental Concepts: Register Transfer, Performing an Arithmetic or Logic operation, Fetching a word from Memory, Storing a word in Memory, Execution of a Complete Instruction: Branch instruction; Multiple-Bus Organization: Hardwired Control: A Complete Processor; Microprogrammed Control: Microinstruction, Microprogram Sequencing, Microinstruction with Next – Address Field.
08 HOURS.


03 HOURS.

Text Books:

Chapter: 1 (expect 1.5, 4.6.6)
2 (expect 2.6.2, 2.11, 2.13)
4 (expect 4.2.6, 4.3, 4.5.3, 4.8)
5 (expect 5.5.2, 5.5.3, 5.5.4, 5.6.3, 5.6.4, 5.8, 5.9.3)
6 (expect 6.5.2, 6.7.3)
7 (expect 7.5.3, 7.5.5, 7.5.6)
9 (Only 9.1, 9.2, 9.3)

Reference Books:
OBJECT ORIENTED PROGRAMMING
LABORATORY WITH C++
(CSE, ISE)

Sub code : CSE47
Exam Marks: 50
Hours/week: 4
IA Marks: 25
Total Hours: 42
Exam Hours: 3

PART – A

1. **Given that an EMPLOYEE Class contains following members:**
   Data members: Employee_Number, Employee_Name, Basic, DA, IT, Net_Sal
   Member functions: to read the data, to calculate Net_Sal and to print data members.
   Write a C++ program to read the data of N employees and computer Net_Sal of each employee (DA = 52% of Basic and Income Tax (IT) + 30% of the gross salary).

2. **Define a STUDENT class with USN, Name, and Marks in 3 tests of subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of two better marks for each student. Print the USN, Name and the average marks of all the student.**

3. **Write a C++ program to create a class called as COMPLEX, and implement the following overloading the function ADD that returns COMPLEX number.**
   (i) ADD (s1, s2) – where a is an integer (real part) and s2 is a COMPLEX number.
   (ii) ADD (s1, s2) – where s1 & s2 are complex numbers.

4. **Write a C++ program to create a class LIST (Linked list) with member functions to insert an element at the front as well as to delete an element from the front of the list. Demonstrate all function after creating a list object.**
5. Write a C++ program to create a template function for quick sort and demonstrate sorting of integers and doubles.

6. Write a C++ program to create a class called as STACK using an array of integers. Implement the following operations by overloading the operators + and –
   (i) s1=s1+element : where s1 is an object of the class STACK and element is an integer to be pushed on the top of the stack.
   (ii) s1 =s1- ; where s1 is an object of the class STACK . – operator pops the element.
   Handle the STACK empty and STACK full conditions. Also Display the contents of the stack after each operation, by Overloading the operator <<.

7. Write a C++ program to create a class called DATE. Accept two valid dates in the form of dd/mm//yy. Implement the following operations by overloading the operators+ and -. After every operation display the result by overloading the operator <<.
   (i) no_of_day = d1 – d2 ; where d1 and d2 are DATE objects, d1>=d2 and , no_of_days is an integer.
   (ii) D2=d1 + no_of_days;where d1 is a DATE object and no_of_days is an integer.

8. Write a C++ program to create a class called as MATRIX using Two dimensional array of integers. Implement the following by overloading the operator == which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction overloading the operators + and – respectively.
   Display the results by overloading the operator <<.
   
   ```
   m3=m1+m2;
   m4=m1-m2;
   ```

   else
     display error

9. Write a C++ program to create a class OCTAL which has the characteristics of an octal number. Implement the following by writing an appropriate constructor and an overloaded operator +.
(i) OCTAL \( h = x; \) - where \( x \) is an integer.
(ii) int \( y = h + k; \) where \( h \) is an OCTAL object and \( K \) is an integer.

Displays the OCTAL result by overloading the operator \( << \). Also display the values of \( h \) and \( y \).

10. Write a C++ program to create a class called QUEUE, with member functions to add an element and to delete and element from the queue. Using these member functions, implement a queue of integers and doubles. Demonstrate the operation by displaying the content of the queue after every operation.

11. Write a C++ program to create a class called DLIST (Doubly linked list) with member functions, to insert a node at a specified position, and delete a node from a specified position of the list. Demonstrate the operation by displaying the content of the list after every operation.

12. Write a C++ program to create a class called STUDENT with data members, USN, Name and age. Using inheritance, create classes UGSTUDENT and PGSTUDENT having fields as semester, fees and stipend. Enter the data for at least 5 student. Find the semester wise average age for all UG and PG students separately.

13. Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator \( << \).
   (i) \( \text{STRING s1 = "VTU",} \)
   (ii) \( \text{STRING s2 = "Belgaum".} \)
   (iii) \( \text{STRING s3 = s1 + s2; - (Use copy / constructor ).} \)

14. Write a C++ program to create a class called BIN_TREE (binary tree) with member functions to perform inorder, preorder, postorder, traversals. create a BIN_TREE object demonstrate traversals.

15. Write a C++ program to create a class called EXPRESSION. Using appropriate member functions convert a given valid INFIX expression into Postfix form. Display the Infix and Postfix expressions.
Note: In the examination questions must be given on lots. Each student must be given one question.

ALGORITHMS LABORATORY
(CSE, ISE)

Sub code: CSE48
Exam Marks: 50
Hours/week: 4
IA Marks: 25
Total Hours: 42
Exam Hours: 3

PART – A

Implement the following using C/C++ Language.

1. Perform recursive Binary search and Linear search. Hence find the time required to search an element.

2. Sort a given set of elements using the Heapsort method.

3. (a) Sort a given set of elements using Merge sort method.
   (b) Check whether a given graph is connected or not using DFS method.

4. Sort a given set of elements using Selection sort and hence find the time required to sort elements.

5. (a) Obtain the Topological ordering of vertices in a given digraph.
   (b) Sort a given set of elements using Insertion sort method.

6. Implement 0/1 Knapsack problem using dynamic programming.

7. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra’s algorithm.

8. Sort a given set of elements using Quick sort method.

10. (a) Print all the nodes reachable from a given starting node in a diagraph using Breadth First Search method.
    (b) Implement All pair shortest paths problem using Floyd’s algorithm.

11. Find a subset of a given set \( S = \{s_1, s_2, \ldots, s_n\} \) of \( n \) positive integers whose sum is equal to a given positive integer \( d \). For example, if \( S = \{1, 2, 5, 6, 8\} \) and \( d = 9 \) there are two solutions \( \{1, 2, 6\} \) and \( \{1, 8\} \). A suitable message is to be displayed if the given problem instance doesn’t have a solution.

12. (a) Implement Horspool algorithm for String Matching.
    (b) Find the Binomial Co-efficient using Dynamic Programming.


14. (a) Print all the nodes reachable from a given starting node in a given digraph using Depth First Search method.
    (b) Compute the transitive closure of a given directed graph using Marshall’s algorithm.

15. Implement N Queen’s problem using Back Tracking.

Note; In the examination questions must be given on lost. Each student must be given one question.