DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW

Course Structure and Syllabus

For

B. TECH
Computer Science and Engineering

2nd Year

as per

NEP-2020

(To be effective from the session 2024-2025)

B. Tech. Computer Science and Engineering

YEAR: SECOND, SEMESTER-III

(To be effective from the session 2024-2025)

S. No.	Subject Code	Subject Name	L-T-P	Evaluation					Credit
				Sessional			ESE	Grand Total	
				CT	TA	Total			
01	NEC-301	Digital Circuits & Logic Design	3-0-0	20	10	30	70	100	3
02	NCS-301	Data Structure	3-1-0	20	10	30	70	100	4
03	NCS-302	Object Oriented Programming	3-1-0	20	10	30	70	100	4
04	NCS-303	Numerical and Statistical Techniques in Computer Science	3-0-0	20	10	30	70	100	3
05	NCS-304	Theory of Automata	3-1-0	20	10	30	70	100	4
06	NCS-305	Emerging Trends in Technology	3-0-0	20	10	30	70	100	3
07	NEC-351	Digital Circuits & Logic Design Lab	0-0-2	-	20	20	30	50	1
08	NCS-351	Data Structure Lab	0-0-2	-	20	20	30	50	1
09	NCS-352	Java Programming Lab	0-0-2	-	20	20	30	50	1
10	NCS-353	Numerical Techniques Lab	0-0-2	-	20	20	30	50	1
11	NCS-354	Theory of Automata Lab	0-0-2	-	20	20	30	50	1
12	NGP-301	General Proficiency		-	50	50	-	50	
Total								**850**	**26**

Note: Admission for Lateral Entry students will be available in Semester -III

B. Tech. Computer Science and Engineering

YEAR: SECOND, SEMESTER-IV

(To be effective from the session 2024-2025)

S. No.	Subject Code	Subject Name	L-T-P	Evaluation					Credit
				Sessional			ESE	Grand Total	
				CT	TA	Total			
01	NCS-401	Design and Analysis of Algorithm	3-1-0	20	10	30	70	100	4
02	NCS-402	Python Programming	3-0-0	20	10	30	70	100	3
03	NCS-403	Computer Organization	3-1-0	20	10	30	70	100	4
04	NEE-401	Network Analysis & Synthesis	3-1-0	20	10	30	70	100	4
05	NEC-404	Fundamentals of Microprocessor	3-0-0	20	10	30	70	100	3
06	NAS-402	Discrete Mathematics	3-0-0	20	10	30	70	100	3
07	AAS-401	Essence of Indian Knowledge Tradition	2-0-0	-	-	-	70	70*	-
08	NCS-451	Design and Analysis of Algorithm Lab	0-0-2	-	20	20	30	50	1
09	NCS-452	Python Programming Lab	0-0-2	-	20	20	30	50	1
10	NCS-453	Computer Organization Lab	0-0-2	-	20	20	30	50	1
11	NEE-451	Network Analysis & Synthesis Lab	0-0-2	-	20	20	30	50	1
12	NEC-455	Microprocessor Lab	0-0-2	-	20	20	30	50	1
13	NGP-401	General Proficiency		-	50	50	-	50	
Total								**850**	**26**

***Marks will not be added in Total (mandatory to Pass)**

Note:

1. After Examination of Semester-IV, it is mandatory for students to undergo a four to six week industrial training.
2. If the student leaves the programme after completing Semester-IV successfully, student will be awarded a **Diploma in Computer Science and Engineering.**

NEC-301

Digital Circuits & Logic Design

L	T	P
3	0	0

Course Outcomes (COs):

After the successful completion of the course, student will be able to:

CO1: Convert different types of codes and number systems in computers and communication, Boolean algebras and Karnaugh maps for reduction of logic expressions.
CO2: Design and analysis of different combinational logic circuits using logic gates.
CO3: Design and analysis of different sequential logic circuits using flipflops.
CO4: Analyze clocked sequential circuits, its state reduction and assignments.
CO5: Design programmable logic devices.

Unit-I 08

Digital system and binary numbers: Number System: Binary, Octal, Hexadecimal, Signed Binary Numbers, Gate-level minimization: Boolean algebra: definition, axioms, Basic Theorems, and Properties, Boolean functions, Canonical and standard forms, NAND and NOR implementation, K-map Method up to Five Variables, don't care conditions, Quine Mc-Clusky method (Tabular method).

Unit-II 08

Combinational logic: Combinational circuits, analysis procedure, design procedure, binary adder, subtractor, Code Converter, BCD adder, decimal adder, magnitude comparator, decoders, encoders, multiplexers, Demultiplexers.

Unit-III 08

Sequential logic: Sequential circuits, storage elements: latches, flip flops, analysis of clocked sequential circuits, state reduction and assignments, design procedure. Registers and Counters: Shift registers, Asynchronous counter, synchronous counter, other counters: Ring and Johnson Counter.

Unit-IV 08

Synchronous and Asynchronous Sequential Circuits: Analysis of clocked sequential circuits with state machine designing, State reduction & assignments, Design procedure. Analysis procedure of Asynchronous sequential circuits, circuit with latches, design procedure, Reduction of state and flow table, Race-free state assignment.

Unit-V 08

Memory and programmable logic: Introduction to Digital Logic families(TTL and CMOS), RAM, ROM, PLA, PAL, Introduction to VHDL, Basics, Design of Combinational and Sequential circuits using VHDL.

Text Books:

1. M. Morris Mano and M. D. Ciletti, "Digital Design", Pearson Education.
2. David J. Comer, "Digital Logic & State Machine Design", Oxford University Press
3. RP Jain, "Modern Digital Electronics", Tata McGraw Hill Publication.
4. Donald P. Leach and Albert Paul Malvino, Digital Principles and Applications, 6th Edition, TMH.

Reference Books:

1. DP Kothari and J.S. Dhillon, "Digital Circuits and Design", Pearson Education
2. A. Anand Kumar, "Fundamentals of Digital Circuits", PHI Learning Pvt. Ltd.

12-03-2024

3. Douglas L. Perry, "VHDL: Programming by Example", McGraw-Hill 4. Jairam Bhaskar, "A VHDL Primer", Prentice Hall PTR.

NCS-301

DATA STRUCTURE

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Students will be able to learn how to represent arrays, linked lists, stacks, queues, trees, and graphs in memory using the algorithms and their common applications.
- Students will able to understanding the concept of recursion, application of recursion and its implementation and removal of recursion.
- Students will be able to learn the computational efficiency of the sorting and searching algorithms.
- Students will be able to learn implementation of Trees and Graphs, and various operations on these data structure.
- Students will capable to identify the alternative implementations of data structures with respect to its performance to solve a real world problem.

Unit-I **08**

Introduction: Basic Terminology, Elementary Data Organization, Built in Data Types, Abstract Data Types. Arrays: Single and Multidimensional Arrays, Representation of Arrays, Derivation of Index Formulae for 1D,2D,3D & nD Array Application of arrays, Sparse Matrices and their representations. Linked lists: Implementation of Singly Linked List using Array, and Pointer, Doubly Linked List, Circularly Linked List, Operations on a Linked List: Insertion, Deletion, Traversal, Polynomial Representation.

Unit-II **08**

Stacks: Basic operations: Push & Pop, Array and Linked List Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Iteration and Recursion-Principles of recursion, Tail recursion, Removal of recursion Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers, and Hanoi towers. Trade-offs between iteration and recursion. **Queues:** Basic operations: Create, Add, Delete, Circular queues, Array and linked list implementation of queues in C, Dequeue and Priority Queue.

Unit-III **08**

Trees: Basic terminology, Binary Trees, Binary Tree Representation: Array and Pointer(Linked List) Representation, Binary Search Tree, Strictly Binary Tree, Complete Binary Tree. Extended Binary Trees, Tree Traversal algorithms: In-order, Pre-order and Post-order, Constructing Binary Tree from given Tree Traversal, Insertion , Deletion, Searching & Modification of data in Binary Search. Threaded Binary trees, Traversing Threaded Binary trees. Huffman coding using Binary Tree. Concept & Basic Operations for AVL Tree, B Tree & Binary Heaps.

Unit-IV **08**

Searching: Sequential search, Index Sequential Search, Binary Search. **Hashing:** Concept of Hashing & Collision resolution Techniques. **Sorting:** Insertion Sort, Selection, Bubble Sort, Quick Sort, Merge Sort, Heap Sort, Radix Sort.

Unit-V **08**

Graphs: Basic terminology, Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.

Text Book:

1. Aaron M. Tenenbaum, YedidyahLangsam and Moshe J. Augenstein "Data Structures Using C and C++" , PHI
2. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education
3. Thareja, "Data Structure Using C" Oxford Higher Education

Reference Books:

1. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publication
2. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education
3. Lipschutz, "Data Structures" Schaum's Outline Series, TMH

NCS-302

OBJECT ORIENTED PROGRAMMING

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand the fundamentals of object oriented programming.
- Understand java programming basics.
- Use inheritance, polymorphism, arrays, threads, and interfaces.

Unit-I **08**

Object-Oriented Analysis: Introduction to Object Oriented Concepts, Object Oriented Analysis Modeling, Data Modeling, Origin of Object-Oriented Design, Object Oriented Design Concepts, Object Oriented Design methods, Class and object definition, Refining operations, Program Components and Interfaces, Annotation for Object-Oriented Design, Implementation of Detail Design.

Unit-II **08**

Java Basic : JAVA environment, JAVA program structure, Tokens, Statements, JVM, Constant and Variables, Data Types, Declaration of variables, Scope of variables, Symbolic constants, Type Casting. Operators: Arithmetic, Relational, Logical assignments, Increment and Decrement, Conditional, Bitwise, Special, Expressions and its evaluation. Object and Class Concept: Defining a Class, Adding variables and Methods to classes, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members, and Nesting of Methods.

Unit-III **08**

Inheritance and Polymorphism: Extending a Class, Overriding Methods, Final Variables and Methods, Final Classes, Finalize Methods. Abstraction: Abstract Methods and Classes, Visibility Control. Interface, Extending Interface, Implementing Interface, Accessing Interface Variable.

Unit-IV **08**

Arrays: One Dimensional and Two Dimensional, Strings, Vectors, Wrapper Classes. Exception Handling: Concepts of Exceptions, Types of Exception, Try and Catch keyword, Nested Try and Catch. Package: System Packages, Using System Package, Adding a Class to a Package, Hiding Classes.

Unit-V **08**

Exception Handling: Concepts of Exceptions, Types of Exception, Try and Catch keyword**. Threads:** Creating Threads, Extending Threads Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, and Synchronization. Input and Output Classes: I/O Streams, The File Class, Byte Stream, Memory Handling, Sequence Input Stream, Object Output Stream, Object Input Stream, Random Access File, Character Stream.

Text Book:

1. E. Balagurusamy, "Programming in Java", TMH Publications.

Reference Books:

1. Peter Norton, "Peter Norton Guide to Java Programming", Techmedia Publications.
2. Naughton, Schildt, "The Complete Reference JAVA 2", TMH.

NCS-303

NUMERICAL AND STATISTICAL TECHNIQUES IN COMPUTER SCIENCE

L	T	P
3	0	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Apply numerical methods to obtain the approximate solutions to the linear and non-linear transcendental equations and find error.
- Identify numerical methods for various mathematical operations and tasks, such as interpolation formulae like forward, backward, and divided difference formulae.
- Apply the appropriate techniques for numerical differentiation and integration problems. Design the numerical solution of initial value problems of the ordinary differential equations with implicit and explicit methods as appropriate
- Analyse of different Statistical Techniques: Moments, Skewness, Kurtosis, Curve fitting, Correlation, Linear, Regression analysis.
- Analysis of Statistical Techniques: Binomial, Poisson and Normal distributions, Sampling theory, Tests of significations: Chi- square test, t-test, and Analysis of variance (one way).

Unit-I **08**

Numbers and their accuracy: Computer Arithmetic, Errors and their Computation, General error formula, Error in a series approximation. Solution of Algebraic and Transcendental Equation: Bisection Method, Iteration method, Method of false position, Newton-Raphson method, Rate of convergence of Iterative methods.

Unit-II **08**

Finite Differences: Difference tables, Polynomial Interpolation: Newton's forward and backward formula, Central Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's. Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula.

Unit-III **08**

Numerical differentiations, Numerical Integration: Trapezoidal rule, Simpson's 1/3 and 3/8 rule, Boole's rule, Waddle's rule. Solution of differential Equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta Methods.

Unit-IV **08**

Statistics: Mean, Mode, Moments, Standard deviation, Skewness, Kurtosis, Curve fitting, Method of least squares, Fitting of straight lines, Polynomials, Exponential curves etc., Correlation, Regression analysis.

Unit-V **08**

Conditional probability and Bayes theorem: Binomial, Poisson and Normal distributions, Sampling theory (small and large), Tests of significations: Chi- square test, t-test, Analysis of variance (one way), Application to engineering, medicine, agriculture etc. Statistical quality control methods, Control charts, $\bar{X}$, R, p, np and c charts.

Text Book:

1. Peter V. O'Neil, Advance Engineering Mathematics Thomson (Cengage) Learning, 2007.
2. J.N. Kanpur, Mathematical Statistics, S. Chand & company Ttd., 2000
3. Jain, Iyengar and Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int
4. Grewal B S, "Numerical methods in Engineering and Science", Khanna Publishers, Delhi

Reference Books:

1. R.K. Jain & S.R.K. Iyenger, Advance Engineering Mathematics, Narosa Publication House, 2002.
2. Chandika Prasad, Advanced Mathematics for Engineers, Prasad Mudralaya, 1996.
3. Sastry S. S, "Introductory Methods of Numerical Analysis", Pearson Education.
4. B. V. Ramana, Higher Engineering Mathematics, Mc Gra Hill Education, 2016.
5. E. Kreysig, Advanced Engineering Mathematics, John Wiley & Sons, 2005.
6. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 2005.
7. S.P. Gupta, Statistical Methods, Sultan and Sons, New Delhi, 2004.
8. Devore, Probability and Statistics, Thomson (Cengage) Learning, 2007.
9. Walpole, Myers, Myers & Ye, Probability and Statistics for Engineers & Scientists, Pearson Education, 2003

NCS-304

THEORY OF AUTOMATA

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understanding the basic terminology of Grammar and construction of logical machine of NFA and DFA with minimization of number of states.
- Learning to generate regular expressions of various languages, its relationship with FA, related theorems and limitation of finite automata.
- Understanding the CFG and its simplification and various forms.
- Able to write description for PDA and understand its relation with CFG
- Basic ability to write simple Turing machines and fair understanding of undecidability

Unit-I **08**

Automata Theory: Introduction to Theory of Computation- Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non-Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore Machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata.

Unit-II **08**

Regular Expressions: Regular Expressions, Transition Graph, Kleen's Theorem, Finite Automata and Regular Expression- Arden's theorem. Regular and Non-Regular Languages: Closure properties of Regular Languages, Pigeonhole Principle, Pumping Lemma, Application of Pumping Lemma, Decidability- Decision properties.

Unit-III **08**

Regular and Non-Regular Grammars: Context Free Grammar(CFG)-Definition, Derivations, Languages, Derivation Trees and Ambiguity, Regular Grammars-Right Linear and Left Linear grammars, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms-Chomsky Normal Form(CNF), Greibach Normal Form (GNF), Chomsky Hierarchy.

Unit-IV **08**

Push Down Automata: Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, Two stack PDA. **Context Free Languages:** Definition, Examples, and properties of CFL: Closure properties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs.

Unit-V **08**

Turing Machines: Basic Turing Machine Model, Representation of Turing Machines, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Universal Turing machine, Linear Bounded Automata, Church's Thesis. **Recursive Function Theory:** Recursive and Recursively Enumerable language, Halting Problem, Post's Correspondence Problem, Introduction to Recursive Function Theory.

Text Book:

1. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia
2. Introduction to languages and the theory of computation, J Martin, 3rdEdition, Tata McGraw Hill

3. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI

Reference Books:

1. Mathematical Foundation of Computer Science, Y.N.Singh, New Age International
2. KLP Mishra and N. Chandrasekaran, "Theory of Computer Science: Automata,Languages and Computation", PHI Learning Private Limited, Delhi India.
3. Peter Linz, "An Introduction to Formal Language and Automata", NarosaPublishinghouse.
4. K. Krithivasan and R. Rama; Introduction to Formal Languages, Automata Theoryand Computation; Pearson Education.

NCS-305

EMERGING TRENDS IN TECHNOLOGY

L	T	P
3	0	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand the basics of information technology.
- Understand and explore IoT applications.
- Understand and explore applications of edge computing and quantum computing.
- Understand the basics of Blockchain technology.
- Critically examine the advantages and disadvantages of future trends in technology.

Unit-I **08**

Information Technology Basics: Introduction need for information storage and processing information technology components, role of information technology information technology and the internet. internet and its tools. internet evolution, basic internet terminology, data over internet, modes of data transmission, types of networks, types of topologies, protocols used in the internet, getting connected to internet applications, internet application and computer ethics.

Unit-II **08**

Internet of Things: What is the Internet of Things? Sensors, their types and features, IoT components: layers, Smart Cities, Industrial Internet of Things

Unit-III **08**

Edge computing: Introduction, advantages, uses, future of the technology. **Quantum computing:** Introduction, advantages, uses, future of the technology.

Unit-IV **08**

Blockchain: -What is Blockchain? Fundamentals, Principles and Technologies, Cryptocurrencies, Smart Contracts, Blockchain Applications and use cases.

Unit-V **08**

Future Trends: Artificial Intelligence, Generative AI, ChatGPT. Brain Computer Interface: Application, Modal and Global Market. Big data analytics, computer vision, natural language processing, machine learning, deep learning, and deep fake technology.

Text Book:

1. Dennis Curtin," Information Technology The Breaking Wave", Tata McGraw Hill.

Reference Books:

1. Somani, A.K., Ramakrishna, S., Chaudhary, A., Choudhary, C., Agarwal."Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics",Springer publications.
2. Saroj Kaushik,"Artificial Intelligence", CENGAGE Learning.

NEC-351

DIGITAL CIRCUITS & LOGIC DESIGN LAB

L	T	P
0	0	2

Course Outcomes (COs):

After the completion of the course, the students are expected to have the ability to:

CO1: Perform verification of gates truth-tables and realization using universal logic gate ICs.

CO2: Design and analyze combinational logic circuits with MUX/DEMUX, decoder, encoder.

CO3: Design & analyze modular sequential circuits with latches, flip-flops.

CO4: Design and simulate different logical circuits using VHDL.

CO5: Design & build mini projects using digital ICs.

LIST OF PRACTICALS

1. Nomenclature of digital ICs, specifications, study of the data sheet, Concept of V_{cc} and ground, verification of the truth tables of logic gates using TTL ICs. (VHDL Simulation)
2. Realization of basic gates using Universal logic gates. (VHDL Simulation)
3. To implement BCD to Excess-3 & vice-versa. (VHDL Simulation)
4. Construction of simple Decoder & Multiplexer circuits using logic gates. (VHDL Simulation)
5. Design of combinational circuit for BCD to decimal conversion to drive 7-segment display using multiplexer. (VHDL Simulation)
6. To implement Adder and Subtractor. (VHDL Simulation)
7. Realization of RS-JK & D flip-flops using Universal logic gates. (VHDL Simulation)
8. Realization of Universal Register using JK flip-flops & logic gates. (VHDL Simulation)
9. Realization of Universal Register using multiplexer & flip-flops. (VHDL Simulation)
10. Construction of Adder circuit using Shift Register & full Adder. (VHDL Simulation)
11. Realization of Asynchronous Up/Down counter. (VHDL Simulation)
12. Realization of Synchronous Up/Down counter. (VHDL Simulation)
13. Implementation of Mini Project using digital integrated circuits and other components

NCS-351

DATA STRUCTURE LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Understand the programming concepts of array, queue, and linked list.
- Understand the role of BFS and DFS.
- Implement linear and binary searching techniques in C programming.
- Implement various sorting techniques in C programming.

LIST OF PRACTICALS

Note: - At least ten experiments are to be conducted from the following list.

1. To implement addition and multiplication of two 2D arrays.
2. To transpose a 2D array.
3. To implement stack using array.
4. To implement stack using linked list.
5. To implement queue using array.
6. To implement queue using linked list.
7. To implement circular queue using array.
8. To implement circular queue using linked list.
9. To implement binary tree using linked list.
10. To implement binary search tree using linked list.
11. To implement tree traversals using linked list.
12. To implement BFS using linked list.
13. To implement DFS using linked list.
14. To implement Linear Search.
15. To implement Binary Search.
16. To implement Bubble Sorting.
17. To implement Selection Sorting.
18. To implement Insertion Sorting.
19. To implement Merge Sorting.
20. To implement Heap Sorting.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-352

JAVA PROGRAMMING LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Write programs in java language.
- Implement interface and package.
- Implement Method Overloading and Method Overriding. Handle exceptions in java.

LIST OF PRACTICALS

Note: - At least ten experiments are to be conducted. Perform practical using JAVA language.

1. Write a program in java which prints your name using command line arguments.

2. Write a program in java which enters three number using command line arguments and print sum and average of the number

3. Write a program to swap the value of 2 variables without using 3rd variable

4. Write a program to calculate the sum of digits of a given integer no.

5. Write a program to compute the sum of the first and last digit of a given number.

6. Write a program in java which enter the number using Data Input Stream and check whether the entered number is even or odd.

7. Write an application that reads a string and determines whether it is a palindrome.

8. Write a program to enter a sentence form keyboard and also find all the words in that sentence with starting character as vowel.

9. Write a Program in java which creates the array of size 5; find the sum and average of the five numbers.

10. Create a java program that has three version of add method which can add two, three, and four integers.

11. Program illustrating Classes and Objects.

12. Program illustrating Method Overloading and Method Overriding.

13. Program illustrating concept of Interface.

14. Program illustrating use of Final and Super keyword.

15. Program that illustrates the Creation of simple package.

16. Program that illustrates the Accessing of a package.

17. Program that illustrates the Handling of predefined exceptions.

18. Program that illustrates the Handling of user defined exceptions.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-353

NUMERICAL TECHNIQUES LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Understand the basic coding concepts of MATLAB.
- Find the root of the Algebraic equations utilizing Bisection Method, Regula – falsi method, and Newton Raphson method in MATLAB
- Implement Newton's Forward Interpolation and Newton's Divided Difference Interpolation
- Implement Numerical Differentiation in MATLAB.
- Implement Numerical Integration using Simpson 1/3 rule and Simpson 3/8 rule.

A. Introduction to MATLAB:

1. Data types and variables
2. Operators
3. Flow control
4. Functions
5. Input / Output
6. Vectors and Matrices
7. M-File

LIST OF PRACTICALS

B. Implementation of Programs in MATLAB: At least ten experiments are to be conducted.

1. WAP to print sum of even and odd numbers from 1 to N numbers.
2. WAP to find the sum of digits of the entered number.
3. WAP to find the eigen values and eigenvectors of a given square matrix.
4. WAP to find the root of the Algebraic equations using Bisection Method.
5. WAP to find the root of the Algebraic equations using Regula - falsi Method.
6. WAP to find the root of the Algebraic equations using Newton Raphson Method.
7. WAP to implement Newton's Forward Interpolation formula.
8. WAP to implement Newton's Divided Difference Interpolation formula.
9. WAP to implement Langranges Interpolation formula.
10. WAP to implement Numerical Integration using Trapezoidal rule.
11. WAP to implement Numerical Integration using Simpson 1/3 rule.
12. WAP to implement Numerical Integration using Simpson 3/8 rule.
13. WAP to implement Numerical Differentiations.

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-354

THEORY OF AUTOMATA LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Able to construct Non-Deterministic Finite Automata (NFA), Deterministic Finite Automata.
- Able to convert NFA to DFA, Minimization of DFA, checking of equivalence of DFAs.
- Able to Implement Mealy and Moore machine
- Able to understand PDA and generation of parse tree for given CFGs.
- Able to build basic understanding of CFG to CNF transformation and single tape TM.

LIST OF PRACTICALS

Note:- Minimum ten experiments are to be performed from the following list.

1. To implement Deterministic Finite Automata
2. To implement Nondeterministic Finite Automata
3. To implement Conversion of NFA to DFA
4. To implement DFA Minimization
5. To implement DFA to regular grammar conversion
6. To implement DFA to regular expression conversion
7. To implement Combining of automata
8. To implement Regular expression to DFA conversion
9. To implement Mealy and Moore machine
10. To implement Pushdown automata
11. To implement Single tape Turing machine
12. To implement Multi-tape Turing machine
13. To implement Context free grammars (CFG) with single symbols
14. To implement CFG with multiple symbols
15. To implement LL Parsing
16. To implement LR Parsing
17. To implement Regular expressions
18. To implement Regular pumping lemma
19. To implement Context free pumping lemma
20. To implement CFG to Chomsky Normal form transformation

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-401

DESIGN AND ANALYSIS OF ALGORITHM

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Implementation of various sorting algorithm and their comparisons.
- Analysis of various problem solved using Divide & Conquer and Greedy techniques
- Implementation of Dynamic Programming concept in solving various problems.

Unit-I **08**

Introduction: Algorithms, analysing algorithms, complexity of algorithms, growth of functions, performance measurements, sorting and order statistics - shell sort, quick sort, merge sort, heap sort, comparison of sorting algorithms, and sorting in linear time.

Unit-II **08**

Advanced Data Structures: Red-Black trees, B – trees, binomial heaps, and Fibonacci heaps.

Unit-III **08**

Design and Analysis Technique: Divide and conquer with examples such as sorting, matrix multiplication, convex hull and searching, greedy methods with examples such as optimal reliability allocation, Knapsack, minimum spanning trees – Prim's and Kruskal's algorithms, single source shortest paths – Dijkstra's and Bellman ford algorithms.

Unit-IV **08**

Dynamic Programming: Knapsack, all pair shortest paths – Floyd-Warshall algorithms, backtracking, branch and bound with examples such as travelling salesman problem, graph colouring, n-Queen problem, and Sum of subsets problems.

Unit-V **08**

Selected Topics: String Matching, theory of NP-completeness, approximation algorithms, and randomized algorithms.

Text Book:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.

Reference Books:

1. RCT Lee, SS Tseng, RC Chang and YT Tsai, "Introduction to the Design and Analysis of Algorithms", McGraw Hill, 2005.
2. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",
3. Berman, Paul," Algorithms", Cengage Learning.
4. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.

NCS-402

PYTHON PROGRAMMING

L	T	P
3	0	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- Understand the basic concept of python.
- Understand the variable, data type, loop, and properties of python.
- Understand the concept of strings and its associated functions.
- Understand the object-oriented concept in python.
- Apply knowledge of python on file using pandas and numpy.

Unit-I **08**

Basics of Python: The programming cycle for python, python IDE, interacting with python programs, elements of python, variables, data types, type conversion. Expressions, assignment statement, arithmetic operators, operator precedence and Boolean expression.

Unit-II **08**

Conditional program execution: Conditional statement, looping, control statements: if, if else, nested if else, for loop, while loop, nested loop. **Lists:** Introduction, properties, accessing list, operations, working with functions and methods. **Tuple**: Introduction, properties, accessing tuple, operations, working with functions and methods. **Dictionaries**: Introduction, properties, accessing values in dictionaries, working with functions and methods.

Unit-III **08**

Strings and Functions: String manipulation: accessing strings, basic operations, string slices. Functions: definition, calling a function, types of functions, function arguments, anonymous functions, global and local variables.

Unit-IV **08**

OOP Concepts: Classes and objects, definition, creating classes, instance methods, new style class, attributes, inheritance, polymorphism, exception classes, custom exception, overloading, overriding and data hiding.

Unit-V **08**

File handling in Python: Opening and closing file, reading and writing files. Searching- Linear and Binary serach, sorting-merge sorting, insertion sort, and selection sort.

Text Book:

1. Allen B. Downey, Think Python: How to think like a Computer Scientist", 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016.
2. Guido van Rossum and Fred L. Drake Jr, "An Introduction to Python- Revised and updated for Python 3.2", Network Theory Ltd, 2011.
3. John V Guttag, "Introduction to computation and programming using Python", Revised and expanded Edition, MIT Press, 2013.

Reference Books:

1. Robert Sedgewick, Kevin Wayne, Robert Dondero, "Introduction to Programming in Python: An Inter-disciplinary Approach", Pearson India Education Services Pvt Ltd, 2016.
2. Timothy A. Budd, "Exploring Python", Mc-Graw Hill Education (India) Private Ltd, 2015
3. Kenneth A. Lambert, "Fundamentals of Python: First Programs", CENGAGE Learning, 2012.

NCS-403

COMPUTER ORGANIZATION

L	T	P
3	1	0

Course Outcomes (COs):

After the successful completion of the course student will be able to:

- The student will conceptualize the basics of organizational and architectural issues of a digital computer.
- The student will learn and perform computer arithmetic operations on integer and real numbers.
- Student will analyse some of the design issues in terms of speed, technology, cost and performance.
- Student will get Exemplified in a better way the I/O and memory organization.

Unit-I **08**

Introduction: Functional units of digital system and their interconnections, System Bus, bus architecture, Bus arbitration. bus and memory transfer. Processor organization, general register organization, stack organization and addressing modes.

Unit-II **08**

Arithmetic and logic unit: IEEE standard for floating point representation, Adder/Subtractor circuit, Look ahead carry adders. Multiplication: Booth's algorithm, Division: Restoring and Non-restoring Method, Arithmetic & logic unit design

Unit-III **08**

Control Unit: Instruction types, formats, instruction cycles and sub-cycles, execution of a complete instruction. Hardwire and microprogrammed control: microprogrammed sequencing, concept of horizontal and vertical microprogramming.

Unit-IV **08**

Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory, organization. ROM memories. Cache memories: Performance, address mapping and replacement, Auxiliary memories, Virtual memory.

Unit-V **08**

Input / Output: Peripheral devices, I/O interface, I/O ports, Isolated and memory mapped I/O, Interrupts: types of interrupts and handling, Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access, I/O channels and processors, standard communication interfaces.

Text Book:

1. William Stalling, "Computer Organization", PHI
2. Vravice,Hamacher & Zaky, "Computer Organization", TMH
3. Mano," Computer System Architecture", PHI

Reference Books:

1. Patterson, Computer Organisation and Design, Elsevier Pub. 2009
2. John P Hays, " Computer Organization", McGraw Hill
3. Tannenbaum," Structured Computer Organization", PHI
4. P Pal chaudhry, ,, Computer Organization & Design", PHI

NEE 301/NEE-401

NETWOKS ANALYSIS AND SYTHESIS

L T P: 3 1 0

Course Outcomes (Cos):

After the completion of the course, students are expected to have the ability to:

CO-1 Apply different network analysis and simplification theorems to dc and ac circuits and verify the solutions using modern tools for lifelong learning

CO-2 Solve network equations using classical methods and verify the solutions using modern tools for lifelong learning

CO-3 Apply Laplace Transformation technique for solution of network equations

CO-4 Calculate two port parameters and analyse network functions to decide stability of networks

CO-5 Define basic terms related with filters and design low pass/high pass passive filters

CO-6 Understand the method to find different type of network function and network function importance

CO-7 Understand different methods use for network synthesis.

UNIT I **08**

Concepts of Circuits: Network element classification, Energy sources, VI relations for R, L and C.

Basic Network Analysis methods: Kirchhoff's laws (KVL & KCL), Mesh analysis for independent, dependent and sinusoidal sources, Super mesh, Nodal analysis for independent, dependent and sinusoidal sources, Super node, Source transformation techniques, duality concept.

Graph Theory: Graph of a Network, definitions, tree, co tree, link, basic loop and basic cut set, Incidence matrix, cut set matrix, Tie set matrix, Loop and Nodal methods of analysis.

UNIT II **07**

Network Theorems (Applications to AC networks): Concept of linearity, and homogeneity Principle, Super-position theorem, Thevenin's theorem, Norton's theorem, maximum power transfer theorem, Reciprocity theorem. Millman's theorem, compensation theorem, Tellegen's theorem.

UNIT III **07**

Frequency domain analysis: Laplace transform solution of Integro-differential equations, transform of waveform synthesized with step ramp, Gate and sinusoidal functions, Initial & final value theorem, Transients in RL, RC&RLC Circuits, initial conditions, time constants. Steady state analysis

UNIT IV **11**

Network function and Two port networks: Concept of complex frequency, Network and Transfer functions for one port and two ports, poles and zeros, Necessary condition for driving point and transfer function, Time response and stability from pole zero plot. Two port parameters – Z, Y,

ABCD, Hybrid parameters, their inverse and image parameters, Relationship between parameters, Interconnection of two ports networks,

UNIT V **07**

Network Synthesis: Positive real function; definition, properties; properties of LC, RC and RL driving point functions, Synthesis of one port networks by Foster's and Cauer methods (forms I and II) synthesis of LC, RC and RL driving-point functions, similarities and dissimilarities between Foster's and Cauer's forms.

Text Books: -

1. A. Chakrabarti, "Circuit Theory" Dhanpat Rai & Co.
2. C.L Wadhwa, "Network Analysis and Synthesis" New Age International Publishers
3. N.C. Jagan and C. Lakshmi Narayana, "Network Analysis" B.S. Publication

Reference Books: -

1. D. Roy Choudhary, "Networks and Systems" Wiley Eastern Ltd.
2. M.E. Van Valkenburg, "Network Analysis", Prentice Hall of India
3. Donald E. Scott: "An Introduction to Circuit analysis: A System Approach" McGraw Hill
4. M.E. Van Valkenburg, "An Introduction to Modern Network Synthesis", Wiley Eastern.
5. Franklin F. Kuo, " Network Analysis and Synthesis" Wiley Second Edition.

11/03/2024

11/03/24

NEC-404

FUNDAMENTALS OF MICROPROCESSOR

L	T	P
3	0	0

Course Outcomes (COs):

At the end of this course students will be able to:

CO1: Learn about microprocessor evolution, architecture, and operation, including addressing modes and interrupts.

CO2: Understand the internal workings of 8085 microprocessors, including registers, ALU, and instruction sets.

CO3: Introductory Knowledge to the architecture and operation of 8086 microprocessors, covering register organization and instruction sets.

CO4: Develop skills in assembly language programming for 8085/8086, covering data transfer, arithmetic, and logic operations.

CO5: Gain expertise in interfacing peripheral devices like DMA controllers, programmable timers, and USARTs with microprocessors.

Unit1: Introduction to Microprocessors: 8

Microprocessor evolution and types, microprocessor architecture and operation of its components, addressing modes, interrupts, data transfer schemes, instruction and data flow, timer and timing diagram. Interfacing devices. Architectural advancement of microprocessors. Typical microprocessor development schemes.

Unit2: 8-bit Microprocessor: 8

Pin diagram and internal architecture of 8085 microprocessor, registers, ALU, Control & statu, interrupt and machine cycle. Instruction sets. Addressing modes. Instruction formats
Instruction Classification: data transfer, arithmetic operations, logical operations, branching operations, machine control and assembler directives.

Unit3: 16-bit Microprocessor: 8

Architecture of 8086 microprocessor: register organization, bus interface unit, execution unit, memory addressing, memory segmentation. Operating modes. Instruction sets, instruction format, Types of instructions. Interrupts: hardware and software interrupts.

Unit4: Introduction to Assembly Language: 8

Assembly language programming based on intel 8085/8086. Instructions: data transfer, arithmetic, logic, branch operations, looping, counting, indexing, programming techniques, counters and time delays, stacks and subroutines, conditional call and return instructions

Unit5: Peripheral Interfacing: 8

Peripheral Devices: 8237 DMA Controller, 8255 programmable peripheral interface, 8253/8254 programmable timer/counter, 8259 programmable interrupt controller, 8251 USART and RS232C.

Books

1. Gaonkar , Ramesh S , "Microprocessor Architecture, Programming and Applications with 8085", Penram International Publishing.
2. Ray A K , Bhurchandi K M , "Advanced Microprocessors and Peripherals", TMH
3. Hall D V ,"Microprocessor Interfacing', TMH
4. Liu and Gibson G A , " Microcomputer System: The 8086/8088 family" ,PHI
5. Aditya P Mathur, " Introduction to Microprocessor", TMH
6. Brey, Barry B, "INTEL Microprocessors", PHI
7. Renu Sigh & B.P.Sigh, "Microprocessor, Interfacing and Applications
8. M Rafiqzzaman, "Microprocessors, Theory and Applications",

NAS - 402

DISCRETE MATHEMATICS

L	T	P
3	0	0

Course Outcomes (COs):

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

- Will be able to apply the computing skills to formulate, solve and analyze interdisciplinary real- world problems for higher study and research.
- Will be able to apply logical skills developed in this course, in various computer applications.
- Will be able to apply the concept of generating functions to solve the recurrence relation.
- Will be able to apply various algebraic structures in different branches of computer science.
- Will be able to identify the main attacks that a computer system can receive, as well as the possible protection and detection methods.

UNIT I (09)

Proposition, Logical connectives, Truth tables, Well formed formula, Tautology, Contradiction, Algebra of proposition, Normal forms, Modus ponens, Modus tollens, Validity. First order predicate, Quantifiers, Inference theory of predicate logic, Proof by implication, converse, inverse, contra-positive, Negation and contradiction, Direct proof, Proof by using truth table, Proof by counter example.

UNIT II (08)

Set Theory: Introduction, Combination of sets, Multi sets, ordered pairs, Set identities. Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions. Combinatorics: Mathematical induction, Basics of counting, Pigeonhole principle.

UNIT- III (08)

Recurrence Relations & Generating function: Recurrence relation of order n with constant coefficients, Homogeneous recurrence relations, Inhomogeneous recurrence relation, Generating function, Closed form expression, Properties of G.F., Solution of recurrence relation using G.F.

UNIT IV (09)

Partially ordered sets and Lattice, Algebraic Structures: Binary composition and its properties, Definition of algebraic structure, Semi group, Monoid, Group, Abelain group, Properties of groups, Permutation group, Sub group, Cyclic group and Integers modulo n.

UNIT V (06)

Elements of coding theory: Introduction, Definitions, Error detecting & correcting code, Harmonic Code and distance, Group, Group (Linear) Codes, Decoding methods. Parity check and Generator matrix, Definition parity check Matrix decoding, Coset decoding.
Hamming's Codes: Concept, implementation as error correcting code, single error correcting (SEC) Code and single error correcting & double error detection code (SEC- DED).

Text Books:

1. Liu and Mohapatra, "Elements of Discrete Mathematics", McGraw Hill
2. Y.N. Singh, "Discrete Mathematical Structures", Wiley India, New Delhi, 2010.
3. R.P. Grimaldi, Discrete and Combinatorial Mathematics, Addison Welsy,
4. S.K. Sarkar, "A Text Book of Discrete Mathematics", S. Chand& Company Ltd., 2012.

Reference Books

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Mc Graw Hill, 2002.
2. J.P. Tremblay & R. Manohar, "Discrete Mathematical Structure with Applications to Computer Science" Mc Graw Hill, 1975.
3. V. Krishnamurthy, "Combinatorics: Theory and Applications", East-West Press.
4. Seymour Lipschutz, M. Lipson, "Discrete Mathematics" Tata Mc Graw Hill, 2005.
5. Kolman, Busby Ross, "Discrete Mathematical Structures", Prentice Hall International.

AAS-401

Essence of Indian Knowledge Tradition

L T P
2 0 0

Course Outcomes:

At the end of this course, students shall be:

- Able to understand Ethos of Indian Tradition
- Able to understand History and Values of Ancient Indian Education System
- Able to exhibit inquisitiveness and draw parallel with Modern International Education System
- Able to exhibit knowledge and understanding of Indian Kings and Kingdom's contribution for India's cause
- Able to understand the concepts of Karma, Vidya, and Darshan

Unit I (8)

Introduction to Indian Knowledge Tradition: Concept of Traditional Knowledge; Rich Indian Heritage of Jnan, Vigynan and Jeevan Darshan; Indian view of "Knowledge"; the 18 Vidyas; Aims and Scope of Traditional Indian Knowledge; Interdisciplinary Ethos of Indian Knowledge Tradition.

Unit II (8)

Intellectual Inquiry into Indian Knowledge Tradition: Ashtavakra (Vedic Civilization); Sushrut (600 BCE); Panini (400 BCE); Chanakya (300 BCE); Aryabhata (5th Century AD); Swami Vivekananda (19th Century AD); MK Gandhi (20th Century AD); Sir VS Naipaul (20th Century AD).

Unit III (8)

King's and Kingdom's Contribution to Indian Knowledge Tradition: Mauryan Dynasty—Nyay, Trade, and Civil Infrastructure; Chola Dynasty—Art, Literature and Architecture; Chalukya Dynasty—Art and Architecture; Gupta Dynasty (Golden Period of Indian Knowledge Tradition)—Art, Mathematics, Astronomy, Physics and Metallurgy; Shivaji Maharaj—Nyay, Culture and Wartime Management.

Unit IV (8)

Traditional Indian Education System—The Gurukul System: Importance of Indian Knowledge System; Perspectives of Indian Knowledge System; Features and Objectives of Gurukul System; Inquisitiveness of Gurukul System; Flaws in Modern Education System; Relevance of Gurukul System in Modern Times; Holistic Education; Indian Thinkers: Buddha, Nagarjuna and Adi Sankara.

Unit V (8)

Modern Indian Philosophy: Valluva: Ahimsa, Morality and Love; Raja Rammohun Roy; J. Krishnamurti; Post-colonialism: Shashi Tharoor, GC Spivak; Osho; PR Sarkar: Progressive Utilization Theory; Rambhadracharya; Sadguru.

Texts Books:

1. Shrimadbhagbadgita,. Geeta Press
2. The Arthashashtra, Kautilya. Penguin Classics, U. K., 1992
3. Chanakya Neeti, Chanakya. Diamond Books, New Delhi, 2020.
4. The Six Systems of Indian Philosophy, F. Max Muller. D. K. Printworld, Delhi

Reference Books:

1. A Sourcebook in Indian Philosophy, Sarvepalli Radhakrishnan, C. A. Moore. Princeton University Press, USA, 1967.
2. Indian Philosophy, Sarvepalli Radhakrishnan. OUP, USA, 2009.
3. My Idea of Education, Swami Vivekananda. Advaita Ashrama, Kolkata.
4. The Ashtavakra Gita, Baij Nath. Kessinger Publishing, 2010, USA.
5. Ashtavakra Samhita, Swami Nityaswarupnanda, Advaita Ashrama, Kolkata.
6. Speech of Hon'ble Mr. Justice Vijender Jain, Chief Justice, Punjab and Haryana High Court, Chandigarh in a seminar of Asia Pacific Jurist Association (APJA) on "Safeguarding the Traditional Knowledge in India" on 28.04.2008 in Delhi.
7. The Indic Knowledge Tradition, Avatans Kumar, in "Beyond Occident", India, TOI, Dec. 12, 2019.
8. Political and cultural history of South India, Neelkanth Shastri.
9. The wonder that was India, AL Bausham.
10. The Cholas: Dynasty of Southern India, Vijaya Ramaswamy.
11. Early Chola Art: Origin and Emergence of Style, Chithra Madhavan.
12. Abhilekhon Mein Aitihasic Bhugola, Jitesh Kumar Singh, Swati Publisher, New Delhi, 2020
13. "A Comparative Study of Attitude of Teachers of Training Colleges and Other Colleges Towards Non-Violence", Dr. Nalina Singh and Dr. Savya Sachi. Journal of the Asiatic Society of Mumbai, Vol. 96, No. 1 (I) 2023.

NCS-451

DESIGN AND ANALYSIS OF ALGORITHM LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Understand the given problem and design an algorithm using various design techniques.
- Implement various algorithms such as sorting, searching, shortest path algorithms etc in C language.
- Analyse the performance of various algorithms.
- Compare different algorithms for solving the same problem.
- Demonstrate the concepts of NP Complete theory by implementation of problems such as: Travelling salesman problem etc.

LIST OF PRACTICALS

Write a program to implement the following-

1. Program for Recursive Binary & Linear Search.
2. Program for Heap Sort.
3. Program for Merge Sort.
4. Program for Selection Sort.
5. Program for Insertion Sort.
6. Program for Quick Sort.
7. Program for Shell Sort.
8. Program to implement Floyd-Warshall's algorithm.
9. Program for sum of subset algorithm.
10. Knapsack Problem using Greedy Solution
11. Perform Travelling Salesman Problem
12. Find Minimum Spanning Tree using Kruskal's Algorithm
13. Implement N Queen Problem using Backtracking

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-452

PYTHON PROGRAMMING LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Understand the fundamentals of Python programming.
- Make programs in Python using list.
- Make programs Python using dictionary.
- Make programs Python using string.
- Make programs Python using tuple.

LIST OF PRACTICALS

Instruction: At least 6 sections are to be implemented.

Section 1: Basic python program

- Python program to print Hello world!
- Python program to add two numbers
- Python program to find the square root
- Python program to calculate the area of a triangle
- Python program to swap two variables

Section 2: Python program on conversion

- Python program to convert kilometres to miles
- Python program to convert Celsius to Fahrenheit
- Python program to convert decimal to binary, octal and hexadecimal
- Python program to find ASCII value of character
- Python program to implement type conversion

Section 3: Basic mathematical program

- Python program to check Armstrong number
- Python program to check if a number is odd or even
- Python program to check leap year
- Python program to find the largest among three numbers
- Python program to check prime number

Section 4: Python program on list

- Python program to check if a list is empty
- Python program to access index of a list using for loop

- Python program to slice list
- Python program to concatenate two lists
- Python program to remove duplicate element from a list

Section 5: Python program on dictionary

- Python program to merge two dictionaries
- Python program to iterate over dictionary using for loop
- Python program to sort a dictionary by value
- Python program to delete an element from a dictionary
- Python program to check if a key is already present in a dictionary

Section 6: Python program on string

- Python program to check if given strings is palindrome or not
- Python program to capitalize the first character of a string
- Python program to compute all the Permutation of the String
- Python program to create a countdown timer
- Python program to count the number of occurrences of a character in string

Section 7: Python program on tuple

- Python program to find the size of a tuple
- Python program for adding a tuple to list and vice-versa
- Python program to sort a list of tuples in increasing order by the last element in each tuple
- Python program to assign frequency to tuples
- Python program to check if any list element is present in tuple

Section 8: Python program on Classes and Objects

- Create a class my class and add some element in it.
- Create a python program to access all elements of a given class
- Create a python program to show OOPs concept
- Create a python program to delete an object in python
- Create a class named Person, use the init() function to assign values for name and age

Section 9 : Python program on files

- Create a python program to make a file
- Create a python program to open and close a given file.
- Create a python program to read and write in file
- Create a python program for copying, moving, and renaming files
- Create a python program for deleting files in python

Section 10: Section 8: Python program on patterns

- Program to print full pyramid using *
- Pascal's triangle pattern using numbers
- Numbered Diamond pattern
- Square pattern in python
- Simple Number triangle pattern

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NCS-453

COMPUTER ORGANIZATION LAB

L	T	P
0	0	2

Course Outcomes (COs):

At the end of this course students will be able to:

- Understand the working of combinational logic circuits.
- Understand the concept of fast adders and their applications.
- Examine the operation of various components of computers.
- Understand the concept of dismantling and assembling of PC.

LIST OF PRACTICALS

1. To design and examine the operations of Half Adder and Full Adder
2. To design and examine the operations of Half subtractor and Full subtractor.
3. To design and examine the operation of 4 bit look ahead carry adder.
4. To design and examine the operations of Registers.
5. To design and examine the operations of Arithmetic Logic Unit (ALU)
6. To design and examine the operations of RAM
7. To study chips, ports, and slots of Motherboard.
8. To study internal architecture and function of Hard Disk Drive.
9. To study internal architecture and function of keyboard.
10. To study dismantling and assembling of PC

Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

NEE – 351/NEE-451

NETWORKS ANALYSIS AND SYNTHESIS LAB

L T P: 0 0 2

COURSE OUTCOMES (COs):

After the completion of the course, students are expected to have the ability to:

CO-1 Validate network theorems
CO-2 Evaluate the time response and frequency response characteristics of RLC series circuit and their resonance conditions
CO-3 Determine Z, Y and ABCD parameters for a given two port network.

LIST OF EXPERIMENTS

Note: - At least ten experiments are to be conducted from the following list.

1. Verification of principle of superposition theorem with AC source.
2. Verification of Thevenin's and Norton's theorem with AC source.
3. Verification of Maximum power transfer theorem in AC circuits.
4. Verification of Tellegen's theorem for two networks of the same topology.
5. Determination of transient response of current in RL and RC circuits with step voltage input.
6. Determination of transient response of current in RLC circuit with step voltage input for under damped, critically damped and over damped cases.
7. Determination of frequency response of current in RLC circuit with sinusoidal AC input.
8. Determination of z and h parameters for a two-port network and compute other parameters.
9. Determination of driving point and transfer functions of a two port ladder network and verify with theoretical values.
10. Determination of image impedance and characteristic impedance of T and Π networks, using O.C. and S.C. tests.
11. Verification of parameter properties in inter-connected two port networks: series, parallel and cascaded.
12. Determination of frequency response of a Twin – T notch filter.
13. To determine attenuation characteristics of a low pass / high pass active filters.

Note: Student may perform above experiments by virtual lab.

14/03/2024

11/03/24

NEC-455

MICROPROCESSOR LAB

L	T	P
0	0	2

Course Outcomes (COs):

After the successful completion of the course student will be able to:

CO1: Learn about 8085 and 8086 microprocessors through hands-on experiments.
CO2: Develop skills in writing assembly programs for various tasks like number manipulation and temperature conversion.
CO3: Acquire the ability to connect peripheral devices like RAM and keyboards to microprocessors.
CO4: Apply lab knowledge to develop practical applications and interface peripherals effectively.

Programming and intyerfacing-based Experiments (any eight)

1. To develop and run a program for finding out the largest/smallest number from a given set of numbers.
2. To develop and run a program for arranging in ascending/descending order of a set of numbers
3. To perform multiplication/division of given numbers
4. To perform conversion of temperature from 0F to 0C and vice-versa
5. To perform computation of square root of a given number
6. To perform floating point mathematical operations (addition, subtraction, multiplication and division)
7. To obtain interfacing of RAM chip to 8085/8086 based system
8. To obtain interfacing of keyboard controller
9. To obtain interfacing of DMA controller
10. To obtain interfacing of PPI
11. To obtain interfacing of UART/USART
12. To perform microprocessor based stepper motor operation through 8085 kit
13. To perform microprocessor based traffic light control
14. To perform microprocessor based temperature control of hot water.

12.03.2024