DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

UNIVERSITY OF LUCKNOW

Course Structure & Syllabus

for

B.Tech. 2nd Year

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE)

(To be effective from the session 2022-2023)

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, artificial intelligence, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

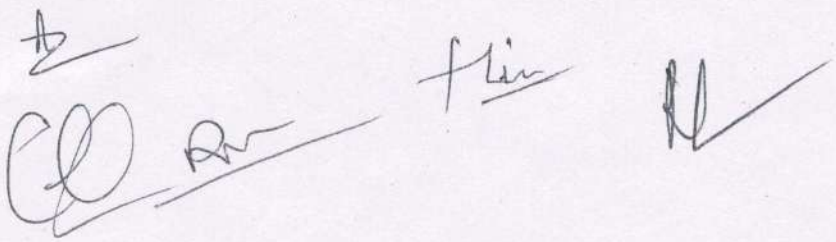
PROGRAM SPECIFIC OUTCOMES (PSOs)

At the end of the program, the student

PSO1: Developing computational systems: Use principles of artificial intelligence, various programming languages, data structures, database management systems, computer algorithms, theory of computation and software engineering for designing and implementing computational systems.

PSO2: Devising networking solutions: Apply the knowledge of systems in the areas related to network technologies, mobile ad hoc and sensor networks, cloud computing, IoT and, information and web security for devising networking solutions.

PSO3: Doing data analytics and designing intelligent systems: Utilize the approaches and tools of artificial intelligence and soft computing, data analytics and machine learning for designing and working with intelligent systems that can extract valuable information from large amount of data and learn from their environment.



Course Structure and Evaluation Scheme for B.Tech. CSE (AI)

YEAR: SECOND, SEMESTER-III

(To be effective from the session 2022-2023)

S. No.	Subject Code	Subject Name	Periods			Evaluation				Grand Total	Credit
						Sessional			ESE		
			L	T	P	CT	TA	Total			
1.	AI - 301	Introduction to Artificial Intelligence	3	1	0	20	10	30	70	100	4
2.	AI - 302	Java Programming	3	1	0	20	10	30	70	100	4
3.	CS - 301	Data Structure Primer using C	3	0	0	20	10	30	70	100	3
4.	CS - 302	Numerical & Statistical Techniques in Computer Science	3	0	0	20	10	30	70	100	3
5.	EC - 301	Digital Circuits & Logic Design	3	0	0	20	10	30	70	100	3
6	AS -302/ AS - 303	Human Values & Ethics / Environment & Ecology	3	0	0	20	10	30	70	100	3
		Practical									
7.	AI - 351	Java Programming Lab	0	0	2	-	20	20	30	50	1
8.	CS - 351	Data Structure Lab	0	0	2	-	20	20	30	50	1
9.	CS - 352	Numerical Technique Lab	0	0	2	-	20	20	30	50	1
10.	EC - 351	Digital Circuits & Logic Design Lab	0	0	2	-	20	20	30	50	1
11.	GP - 301	General Proficiency						50		50	
		Total	**18**	**2**	**8**					**800**	**24**

Abbreviations: CT – Class test **TA – Teacher's Assessment**

ESE – End Semester Examination

Course Structure and Evaluation Scheme for B.Tech. CSE (AI)

YEAR: SECOND, SEMESTER-IV

(To be effective from the session 2022-2023)

S. No.	Subject Code	Subject Name	Periods			Evaluation				Grand Total	Credit
						Sessional			ESE		
			L	T	P	CT	TA	Total			
1.	AS - 404	Discrete Mathematical Structure	3	1	0	20	10	30	70	100	4
2.	CS - 401	Computer Organization	3	1	0	20	10	30	70	100	4
3.	CS - 402	Theory of Automata	3	0	0	20	10	30	70	100	3
4.	AI - 401	Python Programming	3	0	0	20	10	30	70	100	3
5.	EC - 404	Fundamentals of Microprocessor	3	0	0	20	10	30	70	100	3
6.	AS - 402/ AS - 403	Human Values & Ethics/ Environment & Ecology	3	0	0	20	10	30	70	100	3
Practical											
7.	CS - 451	Computer Organization Lab	0	0	2	-	20	20	30	50	1
8.	CS - 452	Automata Lab	0	0	2	-	20	20	30	50	1
9.	AI - 451	Python Programming Lab	0	0	2	-	20	20	30	50	1
10.	EC - 454	Microprocessor Lab	0	0	2	-	20	20	30	50	1
11.	GP - 401	General Proficiency						50		50	
		Total	**18**	**2**	**8**					**800**	**24**

AI-301

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

L	T	P
3	1	0

COURSE OUTCOMES (COs)

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

- Understand how to apply knowledge representation techniques to common AI applications.
- Analyze a problem in hand and do the inference to identify the computing requirements that are essential to solve the problem.
- Understand the concepts related to searching, reasoning and handling uncertainty.
- Understand the concept and type of learning.
- Understand the need and various component of expert system.
- Understand soft computing technologies like fuzzy logic, neural network etc.

Unit-I 08

Introduction: Introduction to artificial intelligence, Scope and applications of AI: Natural language processing, vision, speech recognition, robotics, and expert system. Intelligent agents: Structure and working of intelligent agents.

Unit-II 08

Introduction to search: Searching for solutions, solving state space search, uninformed search strategies, informed search strategies: Depth-first, breadth-first search, heuristic search: hill climbing, best first search, branch and bound.

Unit-III 08

Knowledge Representation: Predicate logic: Unification, modus pones, resolution, dependency directed backtracking. Rule based systems: forward reasoning: conflict resolution, backward reasoning: uses of no backtrack.

Unit-IV 08

Structured knowledge representation: Semantic nets- slots, exceptions and default frames, conceptual dependency, and scripts. **Expert Systems:** Need and justification for expert systems, and knowledge acquisition, and component of an expert system.

Unit-V 08

Handling Uncertainty: Non-monotonic reasoning, probabilistic reasoning, use of certainty factors, and fuzzy logic. Learning: Concept of learning, learning automation, genetic algorithm, learning by inductions, and neural nets.

Text Books:

1. E. Rich and K. Knight, "Artificial intelligence", TMH.
2. N.J. Nilsson, "Principles of AI", Narosa Publ. House.
3. Peter Jackson, "Introduction to Expert Systems", AWP, M.A.

Reference Books:

1. D.W. Patterson, "Introduction to AI and Expert Systems", PHI.
2. R.J. Schalkoff, "Artificial Intelligence - an Engineering Approach", McGraw Hill Int Ed.
3. Charnick, "Introduction to A.I.", Addison Wesley.
4. Marcellous, "Expert System Programming", PHI.
5. Elamie, "Artificial Intelligence", Academic Press.
6. Lioyed, "Foundation of Logic Processing", Springer Verlag.

AI - 302

JAVA PROGRAMMING

L	T	P
3	1	0

Course Outcomes (COs):

- Students will be able to learn about object-oriented programming approach.
- Students will be able to understanding the concept of inheritance, polymorphism, abstraction and encapsulation.
- Students will be able to learn about exceptions and exceptional handling.
- Students will be able to learn about threads and multithreading.
- Students will be able to understand about JDBC, Servlet, AWT, and Swing.

Unit-I **08**

Basic concepts of Object-Oriented Programming: Objects and classes, identifying object relationships, attributes and methods, links and association, generalization and inheritance, aggregation, OMT (object modelling techniques) methodologies, examples and case studies to demonstrate methodologies.

Unit-II **08**

Java Programming Language: Introduction to Java Programming, operators, data type, variable, typecasting. Methods & Classes, object, constructor. **Control Statements:** if-else, switch, for loop, while loop, do-while loop, break and Continue statements. Static keyword and this keyword.

Unit-III **08**

Inheritance: Introduction, types of inheritance (single, multi-level, multiple, hierarchical). **Polymorphism:** method overloading, method overriding, runtime polymorphism, dynamic binding, instanceof operator, and final keyword.

Unit-IV **08**

Abstraction: abstract class, interface. **Encapsulation:** package, access modifiers: public, private, default and protected, encapsulation. Java array and String.

Unit-V **08**

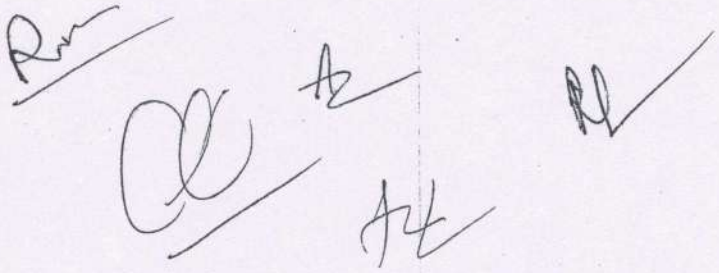
Exception Handling: Exception, try keyword, catch keyword, throw keyword, throws keyword, finally keyword. **Multithreading:** Threads, life cycle of thread, multithreading. Introduction to JDBC, Servlet, AWT and Swing.

Text Books:

1. James Rumbaugh et. al., "Object Oriented Modeling and Design", PHI.
2. Balagurusamy E, "Programming in JAVA", Tata Mcgraw-hill Education Pvt. Ltd.
3. Herbert Schildt, "The Complete Reference: Java" TMH.

Reference Books:

1. Dustin R. Callway, "Inside Servlets", Addison Wesley.
2. Mark Wutica, "Java Enterprise Edition", QUE.
3. Steven Holzner, "Java2 Black book", Wiley Dreamtech Publication.
4. Liang, "Introduction to Java Programming, Comprehensive Version", Pearson Education.
5. Deitel and Deitel, "Java: How to Program" PHI Learning Private Limited, Delhi India.
6. Thampi, "Object Oriented Programming in JAVA" Wiley Dreamtech Publication.



CS-301

DATA STRUCTURE PRIMER USING C

L	T	P
3	0	0

Course Outcomes (COs):

- 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 **07**

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 **09**

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

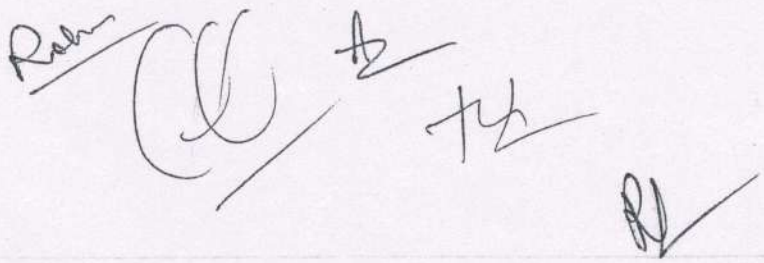
Graphs: Basic terminology, Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: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

Text Books:

1. Seymour Lipschutz, "Data Structures with C", McGraw Hill Education.
2. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein "Data Structures Using C and C++" , PHI.
3. R. Kruse et. al., "Data Structures and Program Design in C", Pearson Education.
4. Thareja, "Data Structure Using C" Oxford Higher Education.

Reference Books:

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



CS - 302

NUMERICAL AND STATISTICAL TECHNIQUES IN COMPUTER SCIENCE

L	T	P
3	0	0

Course Outcomes (COs):
After the completion of the course, students are expected to have the ability to:

- Apply numerical methods to obtain the approximate solutions to the linear and non-linear transcendental and polynomial 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
- Work numerically on the partial differential equations using different methods through of finite difference.

Unit-I: Introduction 08

Numbers and their accuracy, Computer Arithmetic, Mathematical preliminaries, 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, Methods of finding complex roots, Muller's method, Rate of convergence of Iterative methods, Polynomial Equations.

Unit-II: Interpolation 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, Everett's formula.

Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation.

Unit-III: Numerical Integration and Differentiation 08

Introduction, Numerical differentiation Numerical Integration: Trapezoidal rule, Simpson's 1/3 and 3/8 rule, Boole's rule, Waddle's rule.

Unit-IV: Solution of differential Equations 08

Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta Methods, Predictor Corrector Methods, Automatic Error Monitoring and Stability of solution.

Unit-V: Boundary Value problems **08**

Finite difference method, solving eigenvalue problems, polynomial method and power method. Numerical solution of Partial Differential equations. Elliptic, Parabolic and hyperbolic PDEs.

Text Books:

1. Jain, Iyengar and Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int.
2. Grewal B S, "Numerical methods in Engineering and Science", Khanna Publishers, Delhi.

Reference Books

1. Rajaraman V, "Computer Oriented Numerical Methods", Pearson Education.
2. Gerald & Whealey, "Applied Numerical Analyses", AW.
3. T Veerarajan, T Ramachandran, "Theory and Problems in Numerical Methods, TMH.
4. Pradip Niyogi, "Numerical Analysis and Algorithms", TMH.
5. Francis Scheld, " Numerical Analysis", TMH.
6. Sastry S. S, "Introductory Methods of Numerical Analysis", Pearson Education.
7. Gupta C.B., Vijay Gupta, "Introduction to Statistical Methods", Vikas Publishing.
8. Goyal, M, "Computer Based Numerical and Statistical Techniques", Firewall Media, New Delhi.
9. Jaan Kiusalaas, Numerical methods in engineering with MATLAB, Cambridge University Press.
10. C. Woodford and C. Phillips, Numerical methods with worked examples: MATLAB Edition, Springer.

EC-301

DIGITAL CIRCUITS & LOGIC DESIGN

L	T	P
3	0	0

Course outcomes (COs):
The student will be able to

- Gain knowledge between different types of number systems, and their conversions.
- Design various logic gates and simplify Boolean equations.
- Design various flip flops, shift registers and determining outputs.
- Analyze, design and implement combinational logic circuits, e.g. design different types of counters.
- Classify different semiconductor memories.

Unit-I **09**

Digital system and binary numbers: Number System: Binary, Octal, Hexadecimal, Character Codes (BCD, ASCII, EBCDIC) and its arithmetic, signed binary numbers, Cyclic codes, error detecting and correcting codes, Hamming Code.
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 **07**

Combinational logic: Combinational circuits, analysis procedure, design procedure, binary adder-subtractor, 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, ripple counter, synchronous counter, other counters: Johnson & Ring 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, 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, 6thEdition, TMH, 2003.

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

AS – 302/402

HUMAN VALUES AND ETHICS

L	T	P
3	0	0

Course outcomes (COs):

- This course would help to assess ideas about ethics, self-exploration and happiness through reflective enquiry.
- It will aid in evaluating the prevailing problems in society due to differentiation and understanding the importance of human values in relationships.
- The course would lead to knowledge of the ideas of globalization and the world as a nation, for a transformative world order.
- It will help in analyzing ideas of leadership and creativity and using leadership qualities in day-to-day lives.
- It will augment an understanding of cross-cultural ethics and help students learn the art of resolving ethical dilemmas in business.

UNIT 1 **08**

Course Introduction

1. Understanding: Why humans are ethical, why they are not;
2. Understanding the need, basic guidelines, content and process for Value Education;
3. Self Exploration–what is it? – It's content and process;
4. 'Natural Acceptance' and Experiential Validation- as the mechanism for self exploration;
5. Right understanding of Relationship and Physical Facilities- the basic requirements for fulfilment of aspirations of every human being with their correct priority;
6. Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario;
7. Method to fulfil the above human aspirations: understanding and living in **harmony** at various levels

UNIT 2 **08**

Understanding of Human Values and Ethics

1. Understanding the needs of Self ('I') and Body ('Me');
2. Understanding values in human-human relationship;
3. Meaning of Co-existence and Mutual Satisfaction;
4. Understanding Respect;
5. Understanding Comprehensive Human Goals;

UNIT 3 **08**

Effects of Holistic Harmony on Professional Ethics

1. World as a Nation;
2. Definitiveness of Ethical Human Conduct;
3. Basis for Humanistic Education and Humanistic Universal Order;
4. Competence in professional ethics:
   a) Ability to utilize the professional competence for augmenting universal human order;

b) Ability to identify the scope and characteristics of people-friendly and eco-friendly production systems;
c) Ability to identify and develop appropriate technologies and management patterns for above production system;

5. Strategy for transition from the present state to Universal Human Order:
   a) At the level of individual: as socially and ecologically responsible engineers, technologists and managers;
   b) At the level of society: as mutually enriching institutions and organizations;

UNIT 4 **08**

Effects of Holistic Personality for Success

1. Negotiation as a tool for success;
2. Leadership as an attribute of a successful Professional;
3. Managing Stress and Time;
4. Team Building--creating a harmonious environment with apathy to each other;
5. Understanding difference between evolution and revolution;

UNIT 5 **08**

Managing Relationship for Success

1. Understanding and valuing Cross-Cultural Ethics;
2. Managing Relationships (Networking), Personal Effectiveness and Self Leadership;
3. Theory of Constraints;
4. A Decision Making Model: Ethics as making decisions and choices;
5. Conflicts and Ethical Dilemmas;
6. Entrepreneurship and Ethics: A sense of business Ethics;
7. Pragmatic Behaviour of Business to its Colleagues/Competitors

Text Books:

1. Kazuo Ishiguro, 1989, *The Remains of the Day*, Faber and Faber
2. Sussan George, 1976, *How the Other Half Dies*. Penguin Press, Reprint 1991;
3. Amitabh Ghosh, 2008, *Sea of Poppies*. John Murray Publications.

Reference Books

1. B. L. Bajpai, 2004, *Indian Ethos and Modern Management*. New Royal Book Co., Lucknow. Reprinted 2008;
2. R. K. Narayan, 1958, *The Guide*, Viking Press.
3. P. L. Dhar, R. R. Gour, 1990, *Science and Humanism*, Commonwealth Publishers;
4. R. R. Gaur, R. Sangal and G. P. Bagaria, 2010, *A Foundation Course in Human Values and Professional Ethics*, Excel Books.

Relevant movies and documentaries:

1. Story of Stuff (Documentary);
2. The Remains of the Day (Movie);
3. Pursuit of Happyness (Movie);
4. Fences (Movie);
5. Gifted (Movie)

AS – 303/ AS - 403

ENVIRONMENT AND ECOLOGY

L	T	P
3	0	0

COURSE OUTCOMES (COs)

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

- Get the information about environment, ecosystem and also about its functions like Food chain, Ecological pyramids etc.
- Get the complete information about EIA- Environmental Impact Assessment in which the student will get the knowledge about the projects and the process involved in getting the projects.
- Get the knowledge about the different types of resources like land, water, mineral and energy and also about the effects of environment by the usage of these resources. Also get the knowledge about the analysis of polluted water.
- Gain the knowledge about different types of pollution and their treatment techniques like waste water treatment, solid waste management etc.,
- Get the complete information about the all legal aspects of environment protection.

Unit I- Fundamentals of Environment & Ecology 08

Definition, Scope & Importance and Need for public awareness.
Ecosystem- Definition, Energy flow in ecosystem, Ecological succession and Balanced ecosystem.
Effect of human activities on food, Shelter, Economic and social security.
Effect of human activities on environment- Agriculture, Housing, Industry, Mining and Transportation activities.
Basics of Environmental Impact, Assessment and Sustainable development.

Unit II- Natural Resources & Environmental Quality standard 09

Water resources- Availability and quality aspects. Mineral resources, Material Cycle- Carbon, Nitrogen & Sulphur cycles, DO, BOD and COD.
Modern techniques used in analysis of Pollutants- Determination of disinfectants, Pesticides, Ambient Quality standards, Water quality parameters and standards, Turbidity, pH, Suspended solids and hardness,

Unit III- Environmental Pollution & Current Environmental issues 09

Environmental Pollution-Definition, Causes, Effects and control measure of:

1. Air Pollution
2. Water Pollution
3. Soil pollution
4. Marine Pollution

Current environmental issues of importance: Population growth, Climate change & Global warming- effects, Urbanization, Cause of global warming, Acid rain. Ozone layer depletion-

causes and effects on health, Control measures. Photochemical smog, Solid waste management, Waste water treatment.

Unit IV- Energy-Types, Sources and Uses **08**

Different types of energy, Conventional and nonconventional sources- Hydro-electric, Fossil fuel based, Nuclear, Solar, Biomass, Geothermal energy and Biogas. Hydrogen as alternative future source of energy.

Unit V- Environmental protection **06**

Role of Government, Legal aspects, Environment protection Act, Introduction to ISO 14000, Green building concept.

Text Books-

1. Environmental Studies- Dr. D. L. Manjunath, Pearson Education
2. Text book of Environment Science and Engineering- M. Anji Reddy- B S Publication
3. Elements of Environmental Science and Engineering- Dr. P. Meenakshi- Prentice-Hall of India Pvt Ltd, New Delhi, 2008.
4. Environment and Ecology- P.D. Sharma- Rastogi publication 2009.

Reference Books-

1. Principle of Environmental Science and Engineering- P. Venugopalan Rao, Prentice Hall of India.
2. Environmental studies- R. Rajagopalan- Oxford Publication-2005.

AI-351
JAVA PROGRAMMING LAB

L	T	P
0	0	2

LIST OF EXPERIMENTS

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

1. WAP in Java to implement class and object.
2. WAP in Java to illustrate various types of variables in java.
3. WAP in Java to implement constructor.
4. WAP in Java to implement this keyword.
5. WAP in Java for illustrating various forms of Inheritance.
6. WAP in Java to implement polymorphism.
7. WAP in Java for illustrating method overloading.
8. WAP in Java for illustrating method overriding.
9. WAP in Java to implement abstraction using abstract class.
10. WAP in Java to implement abstraction using interface.
11. WAP in Java to illustrate final keyword.
12. WAP in Java to illustrate super keyword.
13. WAP in Java to create a user defined package.
14. WAP in Java to illustrate access modifiers.
15. WAP in Java to implement Encapsulation.
16. WAP in Java to create a thread.
17. WAP in Java to implement multithreading.
18. WAP in Java to illustrate exception handling.
19. WAP in Java to illustrate I/O.

CS-351

DATA STRUCTURE LAB

L	T	P
0	0	2

LIST OF EXPERIMENTS

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.
4. To implement stack using array.
5. To implement stack using linked list.
6. To implement queue using array.
7. To implement queue using linked list.
8. To implement circular queue using array.
9. To implement circular queue using linked list.
10. To implement binary tree using linked list.
11. To implement binary search tree using linked list.
12. To implement tree traversals using linked list.
13. To implement BFS using linked list.
14. To implement DFS using linked list.
15. To implement Linear Search.
16. To implement Binary Search.
17. To implement Bubble Sorting.
18. To implement Selection Sorting.
19. To implement Insertion Sorting.
20. To implement Merge Sorting.
21. To implement Heap Sorting.

CS-352

NUMERICAL TECHNIQUES LAB

L	T	P
0	0	2

Note: - At least ten experiments are to be conducted from section 'B'.

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

B. Implementation of Programs in MATLAB:

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.

EC-351

DIGITAL CIRCUITS & LOGIC DESIGN LAB

L	T	P
0	0	2

LIST OF EXPERIMENTS

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

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

AS - 404

DISCRETE MATHEMATICAL STRUCTURE

L	T	P
3	1	0

Course Outcomes (COs):

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

- Will be able to apply logical skills developed in this course, in various computer applications.
- 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 various algebraic structures in different branches of computer science
- Will be able to apply Graph theoretical concepts to modal, analyze and solve real-world problems.

UNIT-I **08**

Set Theory: Introduction, Combination of sets, Multi sets, ordered pairs, Set identities.

Relations: Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Order of relations.

Functions: Definition, Classification of functions, Operations on functions, Recursively defined functions.

UNIT-II **09**

Propositional Logic: Proposition, Logical connectives, Truth tables, Well formed formula, Tautology, Contradiction, Algebra of proposition, Normal forms, Modus ponens, Modus tollens, Validity.

Predicate Logic: First order predicate, Well formed formula of predicate, Quantifiers, Inference theory of predicate logic.

Notion of Proof: Proof by implication, converse, inverse, contra-positive, Negation and contradiction, Direct proof, Proof by using truth table, Proof by counter example.

UNIT-III **09**

Combinatorics: Mathematical induction, Basics of counting, Pigeonhole principle, Permutations, Combinations, Inclusion-exclusion.

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., Solution of combinatorial problem using G.F.

UNIT-IV **08**

Algebraic Structures: Binary composition and its properties, Definition of algebraic structure, Semi group, Monoid, Group, Abelian group, Properties of groups, Permutation group, Sub group, Cyclic group, Rings and Fields(definition and standard results), and Integers modulo n.

UNIT-V 06

Elements of coding theory: Introduction, Definitions, Error detecting & correcting code, Harmonic Code and distance, Theorems.

Group (Linear) Codes, Decoding methods. Parity check and Generator matrix, Definition parity check Matrix decoding, Coset decoding

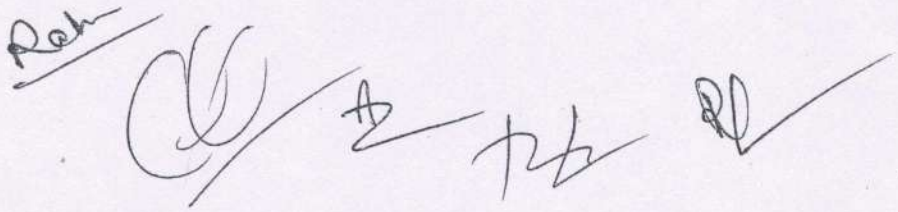
Hamming's Codes: Concept, implementation as error correcting code, single error correcting (SEC) Code and single error correcting & double error detection code (SEC- DED).

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 Wesley.
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.



CS-401

COMPUTER ORGANIZATION

L	T	P
3	1	0

Course Outcomes (COs):

- 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 analyze 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, buses, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer. Processor organization, general register organization, stack organization and addressing modes.

Unit-II **09**

Arithmetic and logic unit: Fixed and floating point representation, IEEE standard for floating point representation, Signed Adder, Subtracter circuits. Look ahead carry adders. Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit design

.

Unit-III **08**

Control Unit: Instruction types, formats, instruction cycles and sub-cycles (fetch and execute etc), micro-operations, execution of a complete instruction. Hardwire and microprogrammed control: microprogramme 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: concept and design issues & performance, address mapping and replacement. Auxiliary memories: magnetic disk, magnetic tape and optical disks. Virtual memory: concept implementation.

Unit-V **07**

Input / Output: Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Serial Communication: Synchronous & asynchronous communication, standard communication interfaces.

Text Books:

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

Reference Books:

1. Patterson, "Computer Organization 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.

CS - 402

THEORY OF AUTOMATA

L	T	P
3	0	0

Course outcomes (COs):
After the completion of course, the 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 **09**

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 **07**

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 **07**

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 09

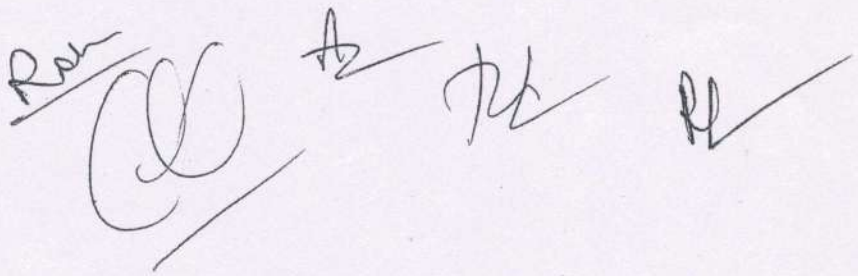
Turing Machines: Basic Turing Machine Model, Representation of Turing Machines, Language Acceptability of Turing Machines, Techniques for Turing Machine Construction, Modifications of Turing Machine, Universal Turing machine, Linear Bounded Automata, Church's Thesis.
Recursive Function Theory: Recursive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

Text Books:

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 Papadimitriou 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", Narosa Publishing house.
4. K. Krithivasan and R. Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.



AI-401

PYTHON PROGRAMMING

L	T	P
3	0	0

Course outcomes (COs):
After the completion of course, the 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 string 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 09

Conditional program execution: Conditional statements, 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: Defining a function, calling a function, types of functions, function arguments, anonymous functions, global and local variables.

Unit-IV 08

OOPS Concepts: Classes and objects, Definition, creating classes, instance method, new style class, attributes, inheritance, polymorphism, exception classes, custom exception, overloading, overriding and data hiding.

Unit-V 07

Working with Data in Python: Printing on screen, reading data from keyboard, opening and closing file, reading and writing files, functions, loading data with pandas and numpy.

Text books:

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.

EC - 404

FUNDAMENTALS OF MICROPROCESSOR

L	T	P
3	0	0

Course Outcomes (COs):

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

- Describe the general architecture of a microcomputer system and architecture & organization of 8085 & 8086 Microprocessor and understand the difference between 8085 and advanced microprocessor.
- Understand and realize the Interfacing of memory & various I/O devices with 8085 microprocessor.
- Understand and classify the instruction set of 8085 microprocessor and distinguish the use of different instructions and apply it in assembly language programming.
- Understand the architecture and operation of Programmable Interface devices and realize the programming & interfacing of it with 8085 microprocessor.

Unit-I **08**

Introduction to Microprocessor: Microprocessor evolution and types, microprocessor architecture and its operation, addressing modes, interrupts, data transfer schemes, instruction and data flow, timer and timing diagram, Basic interfacing concepts, Memory interfacing, Interfacing output displays, Interfacing input devices.

Unit-II **09**

Introduction to 8085 microprocessor: Pin diagram and internal architecture of 8085 microprocessor, registers, ALU, Control & status, interrupt and machine cycle. Instruction sets. Instruction formats. Instruction Classification: data transfer, arithmetic, logic, branch operations, looping, counting, indexing, programming techniques, counters and time delays, stacks and subroutines, conditional call and return instructions.

Unit-III **08**

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

Unit-IV **08**

Introduction to Assembly Language: 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.

Unit-V **07**

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

Text Books:

1. Gaonkar, Ramesh S, "Microprocessor Architecture, Programming and Applications with8085", Penram International Publishing.
2. Hall D V ,"Microprocessor Interfacing', TMH.
3. Liu Y.C. & Gibson G.A., " Microcomputer System: The 8086/8088 family'', Pearson Education.

Reference Books

1. Aditya P Mathur Sigh, "Microprocessor, Interfacing and Applications M Rafiqzzaman, "Microprocessors, Theory and Applications.
2. Ray A K , Bhurchandi K M , "Advanced Microprocessors and Peripherals", TMH.
3. Brey, Barry B, "INTEL Microprocessors", PHI.

CS - 451

COMPUTER ORGANIZATION LAB

L	T	P
0	0	2

LIST OF EXPERIMENTS

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

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

CS - 452

AUTOMATA LAB

L	T	P
0	0	2

LIST OF EXPERIMENTS

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

AI-451

PYTHON PROGRAMMING LAB

L	T	P
0	0	2

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 kilometers 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

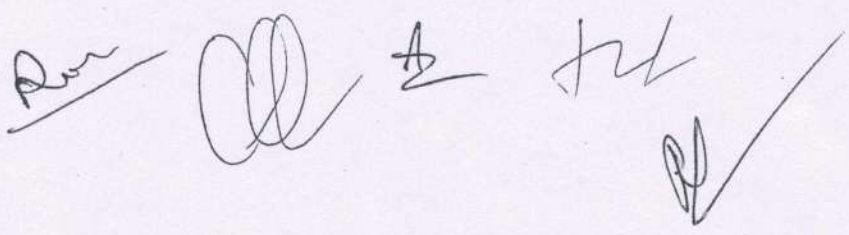
Section 9: Python program on files

- 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 /program, wherever he/she feels in a justified manner.



EC - 454

MICROPROCESSOR LAB

L	T	P
0	0	2

LIST OF EXPERIMENTS

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

Case studies

1. To study 8085 based microprocessor system
2. To study 8086 and 8086A based microprocessor system
3. To study Pentium Processor

Programming based Experiments (any four)

4. To develop and run a program for finding out the largest/smallest number from a given set of numbers.
5. To develop and run a program for arranging in ascending/descending order of a set of numbers
6. To perform multiplication/division of given numbers
7. To perform conversion of temperature from 0F to 0C and vice-versa
8. To perform computation of square root of a given number
9. To perform floating point mathematical operations (addition, subtraction, multiplication and division)

Interfacing based Experiments (any four)

10. To obtain interfacing of RAM chip to 8085/8086 based system
11. To obtain interfacing of keyboard controller
12. To obtain interfacing of DMA controller
13. To obtain interfacing of PPI
14. To obtain interfacing of UART/USART
15. To perform microprocessor based stepper motor operation through 8085 kit
16. To perform microprocessor based traffic light control
17. To perform microprocessor based temperature control of hot water.