DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW

Study & Evaluation Scheme with Syllabus

For

B.Tech. Second Year

(Computer Science and Engineering, Computer Engg. & Information Technology)

On

Choice Based Credit System

(Effective from the Session: 2017-18)

S. No.	Subject Code	Subject Name	L-T-P	ESE Marks	Sessional		Total	Credit
					CT	TA		
1.	RAS301/ ROE030, 032 to 037, 039	Mathematics-III/ Science Based OE	3-1-0	70	20	10	100	4
2.	RVE301/ RAS302	Universal Human Values & Professional Ethics / Environment & Ecology	3-0-0	70	20	10	100	3
3.	REC301	Digital Logic Design	3-0-0	70	20	10	100	3
4.	RCS301	Discrete Structures & Theory of Logic	3-0-0	70	20	10	100	3
5.	RCS302	Computer Organization and Architecture	3-0-0	70	20	10	100	3
6.	RCS305	Data Structures	3-1-0	70	20	10	100	4
7.	REC351	Digital Logic Design Lab	0-0-2	50	30	20	100	1
8.	RCS351	Discrete Structure & Logic Lab	0-0-2	50	30	20	100	1
9.	RCS352	Computer Organization Lab	0-0-2	50	30	20	100	1
10.	RCS355	Data Structures Using C/ Java Lab	0-0-2	50	30	20	100	1
11.	RME101*	Elements of Mechanical Engineering*	3-1-0	70	20	10	100*	--
12.	RCE151*	Computer Aided Engineering Graphics*	0-0-3	50	30	20	100*	--
		TOTAL					**1000**	**24**

CT: Class Test   **TA**: Teacher Assessment   **L/T/P**: Lecture/ Tutorial/ Practical

*B.Tech. IInd year lateral entry students belonging to B.Sc. Stream, shall clear the subjects RCE151/RCE251 and RME101/201 of the first year Engineering Programme along with the second year subjects.

Science Based Open Electives:
a. ROE030/040 Manufacturing Process
b. ROE032/042 Nano Science
c. ROE033/043 Laser System and Application
d. ROE034/044 Space Science
e. ROE035/045 Polymer Science & Technology
f. ROE036/046 Nuclear Science
g. ROE037/047 Material Science
h. ROE039/049 Applied Linear Algebra

S. No.	Subject Code	Subject Name	L-T-P	ESE Marks	Sessional		Total	Credit
					CT	TA		
1.	ROE040, 042 to 047, 049/ RAS401	Science Based OE/ Mathematics-III	3-1-0	70	20	10	100	4
2.	RAS402/ RVE401	Environment & Ecology/ Universal Human Values & Professional Ethics	3-0-0	70	20	10	100	3
3.	REC405	Introduction to Microprocessor	3-0-0	70	20	10	100	3
4.	RCS401	Operating Systems	3-0-0	70	20	10	100	3
5.	RCS402	Software Engineering	3-0-0	70	20	10	100	3
6.	RCS403	Theory of Automata and Formal Languages	3-1-0	70	20	10	100	4
7.	RCS451	Operating Systems Lab	0-0-2	50	30	20	100	1
8.	RCS452	Software Engineering Lab	0-0-2	50	30	20	100	1
9.	RCS453	TAFL Lab	0-0-2	50	30	20	100	1
10.	RCS454	Python Language Programming Lab	0-0-2	50	30	20	100	1
11.	RME201*	Elements of Mechanical Engineering*	3-1-0	70	20	10	100*	--
12.	RCE251*	Computer Aided Engineering Graphics*	0-0-3	50	30	20	100*	--
		TOTAL					**1000**	**24**

CT: Class Test   **TA**: Teacher Assessment   **L/T/P**: Lecture/ Tutorial/ Practical

*B.Tech. IInd year lateral entry students belonging to B.Sc. Stream, shall clear the subjects RCE151/RCE251 and RME101/201 of the first year Engineering Programme along with the second year subjects.

Science Based Open Electives:
   a. ROE030/040 Manufacturing Process
   b. ROE032/042 Nano Science
   c. ROE033/043 Laser System and Application
   d. ROE034/044 Space Science
   e. ROE035/045 Polymer Science & Technology
   f. ROE036/046 Nuclear Science
   g. ROE037/047 Material Science
   h. ROE039/049 Applied Linear Algebra

RCS301: DISCRETE STRUCTURES & THEORY OF LOGIC

UNIT I
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.
Natural Numbers: Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases.

UNIT II
Algebraic Structures: Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphism's, Definition and elementary properties of Rings and Fields, Integers Modulo n.

UNIT III
Partial order sets: Definition, Partial order sets, Combination of partial order sets, Hasse diagram.
Lattices: Definition, Properties of lattices – Bounded, Complemented, Modular and Complete Lattice, Morphisms of lattices.
Boolean Algebra: Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra. Combinational and sequential Circuits.

UNIT IV
Propositional Logic: Proposition, well formed formula, Truth tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference, Natural Deduction.
Predicate Logic: First order predicate, well formed formula of predicate, quantifiers, Inference theory of predicate logic.

UNIT V
Trees: Definition, Binary tree, Binary tree traversal, Binary search tree.
Graphs: Definition and terminology, Representation of graphs, Multi graphs, Bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring.
Recurrence Relation & Generating function: Recursive definition of functions, Recursive algorithms, Method of solving recurrences.
Combinatorics: Introduction, Counting Techniques, Pigeonhole Principle

References:
1. Liu and Mohapatra, "Elements of Discrete Mathematics", McGraw Hill
2. Jean Paul Trembley, R Manohar, "Discrete Mathematical Structures with Application to Computer Science", McGraw-Hill
3. YN Singh, "Discrete Mathematical Structures", Wiley India, New Delhi, First Edition, August 2010.
4. RP Grimaldi, Discrete and Combinatorial Mathematics, Addison Wesley,

5. B. Kolman, R.C. Busby, and SC Ross, "Discrete Mathematical Structures", PHI Learning Private Limited, Delhi India.
6. Norman L. Biggs, "Discrete Mathematics"Oxford Higher Education.
7. Biswal, "Discrete Mathematics and Graph Theory, PHI Learning Private Limited, Delhi India.
8. Goodaire and Parmenter, "Discrete Mathematics with Graph Theory", PHI Learning Private Limited, Delhi India.
9. Lipschutz, "Discrete Mathematics", McGraw Hill
10. Deo N., "Graph Theory with Applications to Engineering and Computer Science", PHI Learning Private Limited, Delhi India

RCS302: COMPUTER ORGANIZATION AND ARCHITECTURE

UNIT I
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, Look ahead carry adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic & logic unit design.

UNIT II
Instruction types, formats, instruction cycles and sub cycles (fetch, execute etc), micro-operations, execution of a complete instruction, Hardwire and micro-programmed control: micro-programme sequencing, concept of horizontal and vertical microprogramming.

UNIT III
Basic concept and hierarchy, semiconductor RAM memories, 2D & $2\frac{1}{2}D$ 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 IV
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.

UNIT V
Architectural Classification Schemes, Flynn's & Feng's Classification, Performance Metrics and Measures, Speedup Performance Laws, Pipelining and Memory Hierarchy Basic and Intermediate Concepts, Linear and Nonlinear Pipeline Processors, Optimization of Cache Performance.

Reference Books:
1. Patterson, "Computer Organization and Design" Elsevier Pub. 2009
2. William Stalling, "Computer Organization", PHI
3. M. Morris Mano, "Computer System Architecture", Pearson Learning
4. Miles Murdocca, Vincent Heuring "Computer Architecture and Organisation: An Integrated Approch" 2nd Edition
5. Kai Hwang, "Advance Computer Architecture", TMH
6. Vravice, Hamacher & Zaky, "Computer Organization", TMH
7. John P Hays, "Computer Organization", McGraw Hill
8. Tannenbaum, "Structured Computer Organization", PHI
9. P Pal Chaudhry, "Computer Organization & Design" PHI
10. Dezso and Sima, "Advanced Computer Architecture", Pearson
11. Alan Clements "Computer Organization and Architecture" , Cengage Learning
12. Behrooz Parhami "Computer Architecture", Oxford

RCS305/ RCS405: DATA STRUCTURES

UNIT I
Introduction: Basic Terminology, Elementary Data Organization, Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big-Oh, Time-Space trade-off.
Abstract Data Types (ADT), Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations.
Linked lists: Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition, Generalized Linked List.

UNIT II
Stacks: Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Tower of Hanoi Problem, Simulating Recursion, Principles of recursion, Tail recursion, Removal of recursion Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.

UNIT III
Trees: Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm.

UNIT IV
Graphs: Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, 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, Introduction to Activity Networks.

UNIT V
Searching: Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting.
Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees .
Hashing: Hash Function, Collision Resolution Strategies.
Storage Management: Garbage Collection and Compaction.

References:
1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI Learning Private Limited, Delhi India
2. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.

3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
4. Thareja, "Data Structure Using C" Oxford Higher Education.
5. AK Sharma, "Data Structure Using C", Pearson Education India.
6. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
7. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.
8. P. S. Deshpandey, "C and Data structure", Wiley Dreamtech Publication.
9. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education
10. Berztiss, AT: Data structures, Theory and Practice, Academic Press.
11. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
12. Adam Drozdek "Data Structures and Algorithm in Java", Cengage Learning

RCS351: DISCRETE STRUCTURE & LOGIC LAB

Understanding of mathematical computation software such as Mapple, Prolog to experiment the followings:

1. Working of Computation software
2. Discover a closed formula for a given recursive sequence vice-versa
3. Recursion and Induction: Practice of proof techniques
4. Practice of various set operations
5. Testing of set operating using software
6. Counting
7. Combinatorial equivalence
8. Permutations and combinations
9. Difference between structures, permutations and sets
10. Implementation of a recursive counting technique
11. N digit binary sequences not having adjacent 1's
12. Probability simulation
13. The Birthday problem
14. Poker Hands problem
15. Baseball best-of-5 series: Experimental probabilities
16. Comparison of theoretical probability with experimental probability
17. Baseball: Binomial Probability
18. Basketball: One and one
19. Expected value problem
20. Binary relations

RCS352: COMPUTER ORGANIZATION LAB

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER and Implementing 4x1 and 8x1 MULTIPLEXERS.
4. Verify the excitation tables of various FLIP-FLOPS.
5. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
6. Design of an 8-bit ARITHMETIC LOGIC UNIT.
7. Design the data path of a computer from its register transfer language description.
8. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
9. Write an algorithm and program to perform matrix multiplication of two n * n matrices on the 2-D mesh SIMD model, Hypercube SIMD Model or multiprocessor system.
10. Study of Scalability for Single board Multi-board, multi-core, multiprocessor using Simulator.

RCS355/RCS455: DATA STRUCTURE USING C/ JAVA LAB

Program in C or C++ for following:

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 queue using array.
5. To implement circular queue using array.
6. To implement stack using linked list.
7. To implement queue using linked list.
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.

REC405: INTRODUCTION TO MICROPROCESSOR

UNIT I
Introduction to Microprocessor, Microprocessor architecture and its operations, Memory, Input & output devices, Logic devices for interfacing, The 8085 MPU, Example of an 8085 based computer, Memory interfacing.

UNIT II
Basic interfacing concepts, Interfacing output displays, Interfacing input devices, Memory mapped I/O, Flow chart symbols, Data Transfer operations, Arithmetic operations, Logic Operations, Branch operation, Writing assembly language programs, Programming techniques: looping, counting and indexing.

UNIT III
Additional data transfer and 16 bit arithmetic instruction, Arithmetic operations related to memory, Logic operation: rotate, compare, counter and time delays, Illustrative program: Hexadecimal counter, zero-to-nine, (module ten) counter, generating pulse waveforms, debugging counter and time delay, Stack, Subroutine, Restart, Conditional call and return instructions, Advance subroutine concepts, The 8085 Interrupts, 8085 vector interrupts.

UNIT IV
Program: BCD-to-Binary conversion, Binary-to-BCD conversion, BCD-to-Seven segment code converter, Binary-to-ASCII and ASCII-to-Binary code conversion, BCD Addition, BCD Subtraction, Introduction to Advance instructions and Application, Multiplication, Subtraction with carry.

UNIT V
8255 Programmable peripheral interface, interfacing keyboard and seven segment display, 8254 (8253) programmable interval timer, 8259A programmable interrupt controller, Direct Memory Access and 8237 DMA controller.
Introduction to 8086 microprocessor: Architecture of 8086 (Pin diagram, Functional block diagram, Register organization).

References:
1. Ramesh Gaonkar, "Microprocessor Architecture, Programming, and Applications with the 8085", 5th Edition, Penram International Publication (India) Pvt. Ltd.
2. Douglas V. Hall, "Microprocessors and Interfacing", Tata McGraw Hill.
3. Yu-cheng Liu, Glenn A. Gibson, "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design", Second Edition, Prentice Hall of India.
4. Barry B. Brey, "The Intel Microprocessors, 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium IV, Architecture, Programming & Interfacing", Eighth Edition, Pearson Prentice Hall, 2009.
5. Peter Abel, "IBM PC Assembly language and programming", Fifth Edition, Prentice Hall of India Pvt. Ltd.
6. Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, "The 8051 Microcontroller and Embedded Systems: Using Assembly and C", Pearson education.

RCS401: OPERATING SYSTEMS

UNIT I
Introduction : Operating system and functions, Classification of Operating systems- Batch, Interactive, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Re-entrant Kernels, Monolithic and Microkernel Systems.

UNIT II
Concurrent Processes: Process Concept, Principle of Concurrency, Producer/ Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.

UNIT III
CPU Scheduling: Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Algorithms, Multiprocessor Scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.

UNIT IV
Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

UNIT V
I/O Management and Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, File system implementation issues, File system protection and security.

References:
1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley
2. Andrew S. Tanenbaum, "Modern Operating System", PHI Learning
3. Tanenbaum /Woodhaull "Operating System Design and Implementation", Pearson Publication.
4. Harvey M Dietel, " An Introduction to Operating System", Pearson Education
5. Flynn, "Understanding Operating System" , Cengage.
6. D M Dhamdhere, "Operating Systems : A Concept basedApproach", McGraw Hill.
7. Charles Crowley, "Operating Systems: A Design-Oriented Approach", Tata McGraw Hill Education".
8. Stuart E. Madnick& John J. *Donovan.Operating Systems.*McGraw Hill.
9. A. K. Sharma, "Operating System", University Press.
10. Achyut S Godbole, Atul kahate , "Operating System", McGraw Hill

RCS402: SOFTWARE ENGINEERING

UNIT I
Introduction: Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes. Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.

UNIT II
Software Requirement Specifications (SRS): Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modelling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS.
Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

UNIT III
Software Design: Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

UNIT IV
Software Testing: Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products.
Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

UNIT V
Software Maintenance and Software Project Management: Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re- Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

References:
1. RS Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Pankaj Jalote, Software Engineering, Wiley
3. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
4. KK Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.

5. Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.
6. Ian Sommerville, Software Engineering, Addison Wesley.
7. Kassem Saleh, "Software Engineering", Cengage Learning.
8. P fleeger, Software Engineering, Macmillan Publication

RCS403: THEORY OF AUTOMATA AND FORMAL LANGUAGES

UNIT I
Introduction; Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill-Nerode Theorem

UNIT II
Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen's Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages . Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

UNIT III
Context free grammar (CFG) and Context Free Languages (CFL): Definition, Examples, Derivation, Derivation trees, Ambiguity in Grammar, Inherent ambiguity, Ambiguous to Unambiguous CFG, Useless symbols, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure proper ties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs.

UNIT IV
Push Down Automata (PDA): Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG, Two stack PDA.

UNIT V
Turing machines (TM): Basic model, definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computerof Integer functions, Universal TM, Church's Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TMs. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory.

References:
1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education.
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. YN Singh "Mathematical Foundation of Computer Science", New Age International.
5. Malviya, AK "Theory of Computation and Application", BPaperback Publications
6. Papadimitrou, C. and Lewis, CL, "Elements of the Theory of Computation", Pearson Publication.

7. K. Krithivasan and R. Rama; Introduction to Formal Languages, Automata Theory and Computation; Pearson Education.
8. Harry R. Lewis and Christos H. Papadimitriou, Elements of the theory of Computation, Second Edition, Prentice-Hall of India Pvt. Ltd.
9. Micheal Sipser, "Introduction of the Theory and Computation", Thomson Learning.
10. Katuri Viswanath, "Introduction to Mathematical Computer Science, An" Universities Press.

RCS451: OPERATING SYSTEMS LAB

1. To implement CPU Scheduling Algorithms
   - FCFS
   - SJF
   - SRTF
   - PRIORITY
   - ROUND ROBIN
2. Simulate all Page Replacement Algorithms
   - FIFO
   - LRU
3. Simulate Paging Technique of Memory Management
Note: The Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

RCS452: SOFTWARE ENGINEERING LAB

For any given case/ problem statement do the following;
1. Prepare a SRS document in line with the IEEE recommended standards.
2. Draw the use case diagram and specify the role of each of the actors. Also state the precondition, post condition and function of each use case.
3. Draw the activity diagram.
4. Identify the classes. Classify them as weak and strong classes and draw the class diagram.
5. Draw the sequence diagram for any two scenarios.
6. Draw the collaboration diagram.
7. Draw the state chart diagram.
8. Draw the component diagram.
9. Perform forward engineering in java. (Model to code conversion)
10. Perform reverse engineering in java. (Code to Model conversion)
11. Draw the deployment diagram.

RCS453: TAFL Lab

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

RCS454: PYTHON LANGUAGE PROGRAMMING LAB

Write a Python program to: -
1. Demonstrate the working of 'id' and 'type' functions
2. To find all prime numbers within a given range.
3. To print 'n terms of Fibonacci series using iteration.
4. To demonstrate use of slicing in string
5.
   a. To add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.
   Sample String : 'abc'
   Expected Result : 'abcing'
   Sample String : 'string'
   Expected Result : 'stringly'
   b. To get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.
6.
   a. To compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.
   b. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.
7. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.
8. To demonstrate use of list & related functions
9. To demonstrate use of Dictionary& related functions
10. To demonstrate use of tuple, set& related functions
11. To implement stack using list
12. To implement queue using list
13. To read and write from a file
14. To copy a file
15. To demonstrate working of classes and objects
16. To demonstrate class method & static method
17. To demonstrate constructors
18. To demonstrate inheritance
19. To demonstrate aggregation/composition
20. To create a small GUI application for insert, update and delete in a table using Oracle as backend and front end for creating form

The lab experiments for this course have to ensure that the following concepts of PYTHON LANGUAGE are covered during lab classes:

Installing Python; basic syntax, interactive shell, editing, saving, and running a script, the concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; reading input from console, writing to console, comments in the program; understanding error messages; Conditions, Boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while);

String manipulations: subscript operator, indexing, slicing a string; other functions on strings: string module, strings and number system, format functions: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers

Lists, tuples, sets, and dictionaries: basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries, Array in Python

Regular Expressions: re modules, match function, search function, modifiers and patterns

Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments. Program structure and design. Recursive functions, scope and global statements, Lambda expressions, Importing Modules, math Module & Random Modules, creating own module.

Exception Handling: Exceptions, except clause, try and finally clause user defined exceptions

File Handling: manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file;

Simple Graphics: "turtle" module; simple 2d drawing - colors, shapes; digital images, image file formats. Graphical user interfaces: event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

Database: cx_ Oracle module, Connections, Executing Queries, calling procedure and functions, Using GUI to access Database.

Object Oriented Programming: Concept of OOP: Abstraction, Encapsulation, Inheritance, and Polymorphism in Python, classes, objects, attributes and methods; defining classes; design with classes, constructors and destructors, inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; aggregation and composition.

Reference books:
1. John M. Sewart, "Python for Scientist", Cambridge Universities Press.
2. Reema Thareja, "Python Programming" Oxford Higher Education.
3. Robert Sedgewick, Kevin Wayne, Robert Dondero, "Introduction to Programming in Python" Pearson
4. Mrak Litz, " Learning Python",O' Reilly
5. Mark Pilgrim, "Dive into Python", Apress
6. James L. Young, "Python made Simple and Practical", Kindle Edition (paperback)
7. Y. Daniel Liang "Introduction to Programming using Python" Pearson