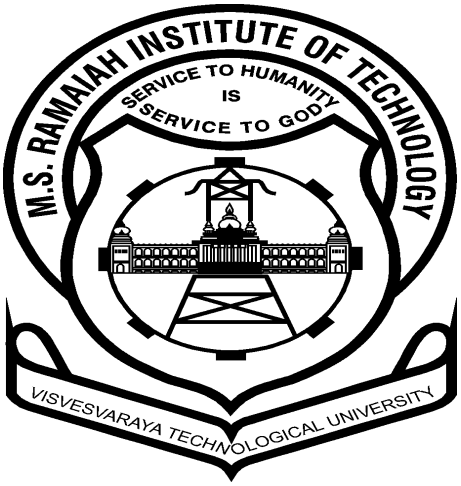
M.S. RAMAIAH INSTITUTE OF TECHNOLOGY

BANGALORE

(Autonomous Institute, Affiliated to VTU)



SYLLABUS

(For the Academic year 2015 – 2016)

III & IV Semester B.E

Information Science and Engineering

Breakup of Credits for BE Degree Curriculum. (I to VIII Semester)

Semester	HSS	BS	ES	PCS	Professional Electives	Other Electives	Project / Seminar/ Internship	Total Credits
I & II	06	20	24					50
III	3	4		18	0	0	0	25
IV		4		21	0	0	0	25
V	2	0		19	4	0	0	25
VI		0		15	8	0	2	25
VII		0		14	4	3	4	25
VIII	1	0			4		20	25
Total	12	28	24	87	20	3	26	**200**

HSS	Humanities and Social Sciences	12
BS	Basic Sciences (Mathematics, Physics, Chemistry)	28
ES	Engineering Sciences (Materials, Workshop, Drawing, Computers).	24
PCS	Professional Core Subjects	87
Prof. Ele	Professional Electives, relevant to the chosen specialization branch	20
Other Ele	Elective Subjects, from other technical and / or emerging subject Areas.	03
Project / Seminar	Project Work, Seminar and / or Internship in industry or elsewhere.	26

M.S. RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 54

(Autonomous Institute, Affiliated to VTU)

SCHEME OF TEACHING FOR THE ACADEMIC YEAR 2015-2016

III SEMESTER B.E. INFORMATION SCIENCE AND ENGINEERING

Sl. No	Subject Code	Subject	Credits*			
			L	T	P	Total
1	ISMAT311	Mathematics-III	3	1	0	04
2	IS332	Computer Organization and Architecture	4	0	0	04
3	IS333	Data Structures	4	0	0	04
4	IS333L	Data Structures Lab	0	0	1	01
5	IS314	Discrete Mathematical Structures	3	1	0	04
6	IS315	Object Oriented Programming	4	0	0	04
7	IS315L	Object Oriented Programming Lab	0	0	1	01
8	IS336	Management & Entrepreneurship	3	0	0	03
		Total	**21**	**2**	**2**	**25**

*** L** : Lecture **T** : Tutorial **P** : Practical

M.S. RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 54

(Autonomous Institute, Affiliated to VTU)

SCHEME OF TEACHING FOR THE ACADEMIC YEAR 2015-2016

IV SEMESTER B.E. INFORMATION SCIENCE AND ENGINEERING

Sl. No	Subject Code	Subject	Credits*			
			L	T	P	Total
1	ISMAT411	Mathematics-IV	3	1	0	04
2	IS432	Data Communications	4	0	0	04
3	IS433	Software Engineering	3	0	0	03
4	IS414	Design & Analysis of Algorithms	4	0	0	04
5	IS414L	Design & Analysis of Algorithms Lab	0	0	1	01
6	IS435	Microprocessors	4	0	0	04
7	IS435L	Microprocessors Lab	0	0	1	01
8	Is416	Finite Automata & Formal Languages	3	1	0	04
		Total	**21**	**2**	**2**	**25**

*** L** : Lecture **T** : Tutorial **P** : Practical

ENGINEERING MATHEMATICS - III

Sub Code : ISMAT311 **Credits : 3:1:0**

Prerequisite: NIL **Contact hours: 42+28**

Course Coordinator: Dr Vijaya Kumar , A V Srivallabha Reddy

Course objectives:

- Learn to solve algebraic, transcendental and ordinary differential equations numerically.
- Learn to fit a curve, correlation, regression for a statistical data.
- Learn to represent a periodic function in terms of sines and cosines.
- Understand the concepts of continuous and discrete integral transforms in the form of Fourier and Z-transforms.
- Learn the concepts of consistency, methods of solution for linear system of equations and eigen value problems.
- Learn the concepts of linear transformation through matrix algebra.

Course Contents

Unit I

Numerical solution of Algebraic and Transcendental equations: Method of false position, Newton - Raphson method.

Numerical solution of Ordinary differential equations: Taylor series method, Euler and modified Euler method, fourth order Runge-Kutta method.

Statistics: Curve fitting by the method of least squares, Fitting a linear curve, fitting a parabola, fitting a Geometric curve**,** Correlation and Regression.

Unit II

Fourier Series: Convergence and divergence of infinite series of positive terms. Periodic functions, Dirchlet conditions, Fourier series of periodic functions of period 2π and arbitrary period, Half range Fourier series, Practical harmonic analysis.

Unit-III

Fourier Transforms: Infinite Fourier transform, Fourier sine and cosine transform, Properties, Inverse transform.

Z-Transforms: Definition, Standard Z-transforms, Single sided and double sided, Linearity property, Damping rule, Shifting property, Initial and final value theorem, Inverse Z-transform, Application of Z-transform to solve difference equations.

Unit IV

Linear Algebra: Elementary transformations on a matrix, Echelon form of a matrix, rank of a matrix, Consistency of system of linear equations, Gauss elimination and Gauss – Seidal method to solve system of linear equations, eigen values and eigen vectors of a matrix, Rayleigh power method to determine the dominant eigen value of a matrix, diagonalization of a matrix, system of ODEs as matrix differential equations

Unit V

Linear Transformations: Introduction to Linear transformations, Composition of matrix transformations, Rotation about the origin, Dilation, Contraction and Reflection, Kernel and Range, Change of basis.

Tutorial:

Problems on Numerical method, curve fitting, Correlation & regression, Fourier series, Z-transforms, Linear algebra, and Linear transformation.

Text Books:

1. Erwin Kreyszig-Advanced Engineering Mathematics-Wiley-India publishers-10th edition-2015.
2. B.S.Grewal - Higher Engineering Mathematics - Khanna Publishers – 42nd edition-2012.
3. Gareth Williams – Linear Algebra with Applications – Jones and Bartlett Press – 6th edition – 2008.

Reference Books:

1. Peter V. O'Neil – Advanced Engineering Mathematics – Thomson Brooks/Cole – 7th edition – 2011.
2. B. V. Ramana – Engineering Mathematics – Tata McGraw Hill Pub. Co. Ltd. – New Delhi – 2008.
3. David C. Lay – Linear Algebra and its Applications – Pearson-3rd edition-2011

Course outcomes:

Students will be able to

CO1. Solve the problems of algebraic, transcendental and ordinary differential equations using numerical methods.(**PO-a,b)**

CO2. Fit a suitable curve by the method of least squares and determine the lines of regression for a set of statistical data.**(PO-a,b)**

CO3. Find the Fourier series expansion of a function in both full range and half range values of the variable and obtaining the various harmonics of the Fourier series expansion for the given numerical data. **(PO-a,b)**

CO4. Find Fourier transforms, Fourier sine and Fourier cosine transforms of functions and solving difference equations using Z-transforms. (**PO-a,b**)

CO5. Find the rank of a matrix, test the consistency and the solution by Gauss elimination and Gauss Siedel iteration methods and find the Kernel and Range of Linear transformations. **(PO-a,b)**

Computer Organization and Architecture

Course code: IS332 **Credits: 4:0:0**

Prerequisites: NIL **Contanct Hours: 56**

Course coordinator(s): Shashidhara H S

Course objectives:

- Introduction to combinational and sequential circuits.
- Study of computer organization, structure and functions through instruction execution, memory and interrupt structures.
- Study of internal structure and functioning of CPU and different categories of CPU based on instruction execution.
- Explore the structure, function and ways of constructing the control unit.
- Introduction to the structure and organization of current computing environment like parallel processing.

Course Contents:

Unit–I

Review of Digital Circuits: Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, Flip-Flops, Registers, Shift Registers, Ripple Counters, Synchronous Counters, Other Counters.

Unit–II

Overview, The Computer System: Organization and Architecture, Structure and Function, Performance Assessment, Computer Components, Instruction Fetch and Execute. Interconnection Structures, Bus Interconnection. Computer Memory System Overview, Cache Memory Principles, Elements of Cache Design, Programmed IO, Interrupt driven IO, Direct Memory Access

Unit–III

The CPU: The Arithmetic and Logic Unit (ALU), Addressing, Instruction Formats, Assembly Language, Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining-Pipelining Strategy, Reduced Instruction Set Architecture, RISC Pipelining

Unit–IV

The Control Unit: Micro-operations, Control of the Processor, Hardwired Implementation, Microprogrammed Control Basic concepts, Microinstruction Sequencing, Microinstruction execution.

Unit–V

Parallel Organization: Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and the MESI protocol, Multithreading and Chip

Multiprocessors, Clusters, Non Uniform Memory Access, Hardware Performance Issues, Software Performance Issues, Multicore Organization

Text Book

1. M. Morris Mano & Michael D. Ciletti, Digital Design, Pearson Education, 5e, 2012.
2. William Stallings, Computer Organization and Architecture, Designing for Performance, 8e, Pearson, 2010.

Reference Books

1. M Murdocca & V Heuring, Computer Architecture and Organization: An integrated Approach, Wiley, 2007.
2. Patterson D. A., Hennessy J. L., Computer Organization and Design, Morgan Kaufmann, 4e, 2011.
3.

Course outcomes:

Students will be able to

CO1. Design combinational and sequential circuits. **(PO-a,b,d)**

CO2. Understand the structure and functions of computer components, instruction execution, interconnection structures, memory and interrupts. **(PO-b,g,l)**

CO3. Solve arithmetic and logic computations involving Integer and floating point numbers. **(PO-a)**

CO4. Design a microprogrammed control unit. **(PO-c)**

CO5. Analyze the performance of various parallel processing methodologies.**(PO-b,l)**

DATA STRUCTURES

Course Code : **IS333** **Credits** : **4:0:0**

Prerequisites : **CS101/CS201** **Contact Hours** : **56**

Course coordinator(s): Deepthi. K

Course Objectives:

- Understand common data structures and be able to implement them
- Understand the importance of data structures and their use in the computer system
- Ability to choose appropriate data structures and algorithms for problem solving.

Course Contents:

Unit- I

Introduction to Data Structures: Definition, Types, Arrays and Structures in C; The Stack: Definition, Representation, Basic operations of stack(PUSH and POP) and its implementation, Applications of stack: Conversion from Infix to Postfix, Evaluation of Postfix expression;

Unit -II

Applications of stack: Recursion: Recursive definition and processes, Examples – Factorial, Tower of Hanoi, Fibonacci numbers; The Queues: Definition, Representation, Primitive operations of queue and its implementation; Circular queues and PRIORITY QUEUES Single- and Double-Ended Priority Queues;

Unit -III

The Linked List: Memory allocation functions; Representation and implementation of operations (Insertion, Deletion and Search) of Singly and Doubly Linked Lists;

Unit-IV

Circular Linked Lists(Insertion, Deletion and Search), Comparing the dynamic and array implementation of lists, Implementation of Header Nodes; Applications of the lists: Adding two polynomials(using Singly list), Checking for Palindrome(using Doubly Linked list), Round Robin Scheduling;

Unit- V

Trees: Binary Trees, Binary Tree Representations, Representing Lists as Binary trees, Trees and their applications; Threaded Binary Trees; Binary Search Tree: Creation and Deletion;

Text Books

1. Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C", 2nd Edition, PHI, 2009.

Reference Books

1. Horowitz and Sahani. "Fundamentals of Data Structures", 2nd Edition, Galgotia Publication Pvt Ltd.,New Delhi, 2011
2. Behrouz A. Forouzan and Richard F. Gilberg,"Computer Science A Structured Programming Approach using C", Second Edition, Thomson Publications,2007.
3. R. Kruse, "Data Structures and Program Design in C", Pearson Education, 2nd Edition, 2009.

Course outcomes:

Students will be able to

CO1. Understand the concept of data structures and implement stack data structures and its applications. **(PO-a,b,c)**

CO2. Understand the working of queues, its types, applications and their implementations. **(PO-a,b,c)**

CO3. Understand, develop and implement Singly and Doubly linked lists using dynamic memory allocation. **(PO-a,b,c)**

CO4. Understand, develop and implement Circular linked lists using dynamic memory allocation. **(PO-a,b,c)**

CO5. Understand the concept of non linear structures, their design and their implementation. **(PO-a,b,c)**

Data Structures Lab

Course Code : **IS333L** **Credits** : **0:0:1**

Prerequisites: **CS101/CS201** **Contact Hours: 28**

Course coordinator(s): Koushik S

Course objectives:

- Design common data structures and be able to implement them.
- Understand the importance of data structures and their use in the computer system.
- Ability to choose appropriate data structures and algorithms for problem solving.

Course Contents:

1. Conversion of a valid Infix expression to Postfix Expression using stacks.
2. Evaluation of a valid Postfix expression using stacks.
3. Recursion
    a. Tower of Hanoi problem for n disks using recursion.
    b. Binary search and GCD using recursion.
4. Linear queue and its primitive operations with supportive display() function.
5. Circular queue and its operations with supportive display() function.
6. Singly Linked List with the following operations:
    a. Inserting a node(Beginning, End & Given any desired position)
    b. Deleting a node (Beginning, End & Given any desired position)
    c. Display
7. Circular Linked List with the following operations:
    a. Inserting a node(Beginning, End & Given any desired position)
    b. Deleting a node (Beginning, End & Given any desired position)
    c. Display
8. Doubly Linked List with the following operations:
    a. Inserting a node(Beginning, End & Given any desired position)
    b. Deleting a node (Beginning, End & Given any desired position)
    c. Display
9. To insert a given element into an ordered doubly linked list.
10. Binary tree operations:
    a. Creation
    b. Traversal(Inorder, Preorder and Postorder)
11. Creation of Binary Search tree.
12. Create an expression tree and evaluate it.

Text Book:

2. Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C".
3. Michael J. Folk, Bill Zoellick and Greg Riccardi, "File Structures-An Object Oriented Approach with C++".

Reference Books:

4. Horowitz and Sahani. "Fundamentals of Data Structures", Galgotia Publication Pvt Ltd., New Delhi.
5. Behrouz A. Forouzan and Richard F. Gilberg,"Computer Science A Structured Programming Approach using C", Second Edition, Thomson Publications.
6. R. Kruse, "Data Structures and Program Design in C", Pearson Education.

Course outcomes:

CO1. Design and implement the concepts of data structures. **(PO-a,b,c,d)**

CO2. Appreciate the usage of data structures in various domains. **(PO-a,b)**

CO3. Produce a substantial written documentation. **(PO-j)**

DISCRETE MATHEMATICAL STRUCTURES

Course Code : **IS314** **Credits** : **3:1:0**

Prerequisites: **Nil** **Contact Hours** : **42 + 28**

Course coordinator(s): Pratima. M. N

Course objectives:

- Perform set operations and also to solve logical reasoning to verify the correctness of the logical statements.
- Understand the properties of relations and its importance in mathematics, computers and several other applications
- Apply the properties of relations and find the partially ordered sets and lattices.
- Highlight the concepts of graphs and its usefulness in computing applications.
- Understand the need of mathematical structures and techniques by introducing computing applications.

Course Contents:

Unit-I

Fundamentals: Sets and subsets, operations on sets, Sequences. Logic: Propositions and Logical Operations, Conditional statements, Methods of proof, Mathematical Induction.

Unit-II

Counting: Permutations and combinations, Pigeonhole Principle, Recurrence relations. Relations and Digraphs: Product sets and partitions, relations and digraphs, paths in relations and digraphs, properties of relations, equivalence relations, operations on relations, transitive closure and Warshall's algorithm.

Unit-III

Functions, Functions for computer science, permutation functions, order relations and structures: partially ordered sets, extremal elements of partially ordered sets, lattices.

Unit-IV

Graphs and graph models, graph terminology and special types of graphs, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths, Binary operations revisited, semigroups.

Unit-V

Groups, other mathematical structures (rings, fields and Fermets little theorem). Coding of binary information and error detection.

Tutorial:

Problems on Set theory, Logic, Permutations and combinations, Pigeonhole principle, Relations, Functions, Partial order relations, Graph theory, Binary operations , Groups, Semi groups, Coding

Text Books:

1. Bernard Kolman, Robert C. Bushy, Sharon Cutler Ross, Discrete Mathematical Structures, 6 th edition, PHI(all topics except graphs).
2. Kenneth H Rosen, Discrete Mathematics and its applications, 6th Edition, Tata McGraw-Hill.

References:

1. Ralph P.Grimaldi, B.V Ramana ,Discrete and Combinatorial Mathematics, Fifth edition.
2. J.P.Trembly, R. Manohar, Discrete mathematical structures with applications to Computer Science , McGraw Hill.
3. Richard Johnsonbaugh, Discrete Mathematics, Pearson Education Asia.

Course Outcomes

The student is able to

CO1: Apply the properties of set theory on proof of statements and also to solve logical reasoning to verify the correctness of the programs. **(PO-a,b)**

CO2: Apply the properties of relations on real world examples. **(PO-a,b)**

CO3: Develop structures useful in set theory, algebra by the study of partially ordered sets and lattices. **(PO-a,b)**

CO4: Apply the concepts of graphs in computers and several other applications. **(PO-a,c)**

CO5: Analyze the different mathematical structures and techniques by introducing computing applications. **(PO-a,b)**

OBJECT ORIENTED PROGRAMMING

Course Code : **IS315** **Credits** : **4:0:0**

Prerequisites: **CS101/CS201** **Contact Hours : 56**

Course coordinator(s): T.Tamilarasi

Course Objectives

- Explain the need of using Object Oriented Programming in the real world applications.
- Describe the OOPs, terminology and Structure.
- Differentiate OOPs systems with procedural systems.
- Design programs using classes and objects for C++.
- Specifying mechanism of deriving a new class from older classes through inheritance.
- Construct applications to provide flexible options for the creation of new definitions for some of the operators.
- Implement methods to select appropriate member function during run time.
- Design a program using Templates and standard template libraries, Exception Handling

Course Contents

Unit-I

Introduction : A Review of structures, Procedure- oriented programming systems, object oriented programming systems, reference variables, Function overloading , Default values for formal argument, Classes and objects: Introduction to classes and objects, Member function and member data, Objects and Functions, Objects and Arrays, Namespaces.

Unit-II

Dynamic memory management : Dynamic memory allocation, Dynamic memory deallocation, Constructors and destructors , Inheritance : Introduction, base class

and derived class pointers, function overriding, base class initialization, Protected access specifiers , Different kind of inheritance .

Unit-III

Operator overloading , overloading various operators: overloading increment and decrement , overloading unary minus and unary plus operator, overloading Arithmetic operators, relational and Assignment operators , type conversion.

UML notation- for classes, objects, Generalization, Associations and polymorphism and relationships

Unit-IV

Virtual Functions and dynamic polymorphism: Need, Virtual Functions, Mechanism, Pure Virtual Functions, Virtual Destructors. Streams, class hierarchy and file handling: streams, class hierarchy for handling functions, Binary output/Input files, Opening and closing files, File pointers and manipulators

Unit-V

Templates: Introduction, function templates, class templates and standard template libraries, Exception Handling: Introduction, C-Style Handling of Error-generating Code, C++ style solution, Limitation of exception handling.

Text books

1. Sourav Sahay, Object Oriented Programming Using C++ ,Sourav Sahay, 2nd edition 2013
2. Mark priestley, Practical object oriented design with UML, Tata McGraw-Hill, 2nd edition 2005

References

1. Herbert Schildt, The Complete Reference C++, 4th Edition, Tata McGraw-Hill, 2005.

Course Outcomes

Student will be able to

CO1: Understand the need of using Object Oriented Programming in the real world applications using classes and objects. **(PO-a,b,l)**

CO2: Understanding the mechanism of deriving a new class from older classes through inheritance. **(PO-a,c,e)**

CO3: Constructing applications to provide flexible options for the creation of new definitions for some of the operators. **(PO-a,b,c,e)**

CO4: Building of programs for automatic initialization of objects and destroy objects that are no longer required. **(PO-a,b,c,e)**

CO5. Designing a program using Templates & Exception Handling. **(PO-a,b,e,l)**

OBJECT ORIENTED PROGRAMMING LAB

Course Code : **IS315L** **Credits**: **0:0:1**

Prerequisites: **CS101/CS201** **Contact Hours : 28**

Course coordinator(s): Dayananda.P

Course Objectives:

- Design programs using functions, array of structures.
- Build programs for automatic initialization of objects and destroy objects that are no longer required through constructors and destructors.
- Specifying mechanism of deriving a new class from older classes through inheritance
- Construct applications to provide flexible options for the creation of new definitions for some of the operators.
- Implement methods to select appropriate member function during run time.
- Design a program using Templates and standard template libraries, Exception Handling

Course Contents:

1) a. Write a C++ program to illustrate the concept of inline functions
   b. Write a C++ program to illustrate the use of default arguments
2) Write a C++ program to demonstrate the use of function and array of structure
3) Write a C++ program to demonstrate the use of friend function.
4) Write a C++ program to demonstrate the concept of classes and constructors.
5) Program to illustrate the concept of different types of inheritance using derived class and base class.
6) Write a program to illustrate the concept of derived class constructor in inheritance.
7) Write a C++ program to illustrate the concept of dynamic memory allocation, constructor, destructor, copy constructor and the use of destructor to destroy the memory space.
8) Write a C++ program to Implement the operations by overloading the binary operators and display the results by overloading the operator <<.
9) Write a C++ program to Implement the overloading of operators unary and binary. Display the results by overloading the operator <<.
10) a. Write C++ program to use try catch statements to handle division by zero and out-of-bounds exception.
    b. Write a C++ program to illustrate the template function.
11) Write a C++ program to illustrate the class template.
12) Write a C++ Program to Illustrate the concept of exception handling.

Reference books

1) Sourav Sahay , Object Oriented Programming with C++,. Oxford University Press, 2006
2) E Balaguruswamy, Object Oriented Programming Using C++ ,Tata McGraw-Hill, 4th edition 2008
3) Herbert Schildt, The Complete Reference C++, 4th Edition, Tata McGraw-Hill, 2005.

Course outcomes:

Students will be able to

CO1. Design programs using classes and objects for C++**. (PO-a,b,k)**
CO2. Building of programs for automatic initialization of objects and destroy objects that are no longer required. **(PO-a,e,g,k)**
CO3. Specifying mechanism of deriving a new class from older classes through Inheritance. **(PO-b,c,e)**
CO4. Constructing applications to provide flexible options for the creation of new definitions for some of the operators. **(PO-a,b,k)**
CO5. Designing a program using Templates & Exception Handling. **(PO-a,j,k)**

MANAGEMENT AND ENTREPRENEURSHIP

Course Code : **IS336** **Credits** : **3:0:0**

Prerequisites: **NIL** **Contact Hours : 42**

Course coordinator(s): Myna.A.N.

Course Objectives

- Provide an insight into management and various approaches used
- Describe the various functionalities of management
- Provide the knowledge and skills required to become an entrepreneur
- Discuss the process of setting up a small business

UNIT-I

MANAGEMENT: Nature and Functions of Management-Importance, Definition, Functions, Levels, Organisational Functions, Roles of a Senior Manager, Managerial Skills, Managerial Effectiveness, Management and Administration, Development of Management Thought – Early Classical approaches-Scientific Management, Administrative Management, Bureaucracy, Neo Classical Approaches- The Human Relations Movement, Modern Approaches- Systems Approach

UNIT II

PLANNING: Nature, Importance, Types of plans (Definitions and Meaning only), Steps, DECISION MAKING: Meaning, Types, Steps, ORGANISATION: Meaning, Characteristics, Typology, Process, Principles, COORDINATION: Distinction between coordination and cooperation, Distinction between Coordination and Control, Need for Coordination, Requisites for Excellent Coordination, Types of Coordination, Techniques

UNIT III

DIRECTION AND SUPERVISION: Requirements of Effective Direction, Giving Orders, Motivation-Meaning, Nature, Motivation- Meaning, Nature, Motivation Theories-Maslow's Theory, Herzberg's Theory, McClelland's Need for Achievement Theory, Japanese Model of Motivation-Theory Z, COMMUNICATION: Importance, Purposes, Formal Communication, Forms of Communication, Informal, Communication Process, Barriers to Communication

UNIT-IV

LEADERSHIP: Difference between a Leader and a Manager, Characteristics of Leadership, Functions of a Leader, Traditional Approaches to Leadership- Traits

Approach, Behavioural Approach, MANAGERIAL CONTROL: Steps in a Control Process, Need for Control System, Benefits of Control, Essentials of Effective Control System

UNIT-V

ENTREPRENEURSHIP: Importance, Concepts, Characteristics of a Successful Entrepreneur, Classification of Entrepreneurs-Based on Functional Characteristics, Based on the Developmental Angle, Based on types of Entrepreneurial Business, SETTING UP A SMALL BUSINESS: Formalities of Setting a Small Business Enterprise – Flowchart, Selection of Project, Product of Service Selection, Project Feasibility Study, Business Plan Preparation. Registration, Project Report Preparation, Implement the Project and obtain Final Clearances.

Text Books:

1. P.C. Tripathi, P.N.Reddy, Principles of Management, 5th Edition, Tata McGraw-Hill, 2012
2. Poornima M Charanthimath, Entrepreneurship Development Small business enterprises, Pearson Education, 2008

Reference Books:

1. Ramesh B Rudani, Principles of Management, Tata McGrawHill, 2013
2. Robert Lusier, Management Fundamentals – Concepts, Application, Skill Development, 5th Edition, Cengage Learning, 2012
3. S.S. Khanka, Entrepreneurial Development, S. Chand & Company Limited, 2012, ISBN 10: 8121918014 / ISBN 13: 9788121918015

Course Outcomes:

The students will be able to

CO1: Analyze the various approaches of management. **(PO-h)**

CO2: Apply the administrative skills of planning, decision making and coordinating. **(PO-h)**

CO3: Analyze motivation theories and apply them to direct and supervise employees.

PO-h,I,j)

CO4: Apply effective communication and leadership skills. **(PO-h,I,j)**

CO5: Identify the skills required to become a successful entrepreneur. **(PO-h,j)**

ENGINEERING MATHEMATICS - IV

Sub Code : ISMAT411 **Credits : 3:1:0**

Prerequisite : NIL **Contact Hours : 42+28**

Course Coordinators: Dr Vijaya Kumar ,
A V Srivallabha Reddy

Course objectives:

- Learn the concepts of finite differences, interpolation and it applications.
- Understand the concepts of PDE and its applications to engineering.
- Learn the concepts of Random variables and probability distributions.
- Learn the concepts of probability distributions involving two random variables.
- Learn the concepts of stochastic process, Markov chain and queuing theory.
- Construct the various tests essentially needed for the testing of small samples for the testing of hypothesis.

Course Contents:

Unit I

Finite Differences and Interpolation: Forward, Backward differences, Interpolation, Newton-Gregory Forward and Backward Interpolation, formulae, Lagrange interpolation formula and Newton divided difference interpolation formula (no proof).

Numerical Differentiation and Numerical Integration: Derivatives using Newton-Gregory forward and backward interpolation formulae, Newton-Cotes quadrature formula, Trapezoidal rule, Simpson 1/3rd rule, Simpson 3/8th rule.

Partial Differential Equations - I: Introduction to PDE, Solution of PDE – Direct integration, Method of separation of variables.

Unit II

Random Variables: Random Variables (Discrete and Continuous), Probability density function, Cumulative distribution function, Mean, Variance, Moment generating function..

Probability Distributions: Binomial distribution, Poisson distribution, Normal distribution, Exponential distribution and Uniform distribution.

Unit III

Joint probability distribution: Joint probability distribution (both discrete and continuous), Conditional expectation, Simulation of random variable.

Stochastic Processes: Introduction, Classification of stochastic processes, Discrete time processes, Stationary, Ergodicity, Autocorrelation, Power spectral density.

Unit IV

Markov Chain: Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states, Markov and Poisson processes.

Queuing theory: Introduction, Concepts and M/G/1 and M/M/1 queuing systems with numerical illustration.

Unit-V

Sampling Theory : Sampling, Sampling distributions, Standard error, Weak law of large numbers(without proof), Central limit theorem, Test of Hypothesis for means, Confidence limits for means, Student's t-distribution, F-distribution, Chi-Square distribution as a test of goodness of fit.

Tutorial:

Problems on Interpolation, Differentiation, Integration, Random variables, Probability distribution, Stationary Process, Markov chain, Queuing models, t-test, and Chi-square test.

Text Books :

1. Erwin Kreyszig - Advanced Engineering Mathematics-Wiley-India publishers-10th edition-2015.
2. B.S.Grewal - Higher Engineering Mathematics - Khanna Publishers - 40th edition-2007.
3. R.E. Walpole, R. H. Myers, R. S. L. Myers and K. Ye – Probability and Statistics for Engineers and Scientists – Pearson Education – Delhi – 8th edition – 2007.

Reference Books :

1. Sheldon M. Ross – Probability models for Computer Science – Academic Press – 2009.
2. Murray R Spiegel, John Schiller & R. Alu Srinivasan – Probability and Statistics – Schaum's outlines -2nd edition.
3. Kishor S. Trivedi – Probability & Statistics with reliability, Queuing and Computer Science Applications – PHI – 2nd edition – 2002.

Course outcomes:

Students will be able to

CO1. Find a polynomial from the given data for estimation, finding extreme values of a function, radius of curvature, arc length, surface area etc. using numerical differentiation and integration. **(PO-a,b)**

CO2. Find solution of partial differential equations by direct integration method and separation of variables. **(PO-a,b)**

CO3. Express the probability distribution arising in the study of engineering problems and their applications. **(PO-a,b)**

CO4. Apply the stochastic process and Markov Chain in prediction of future events. **(PO-a,b)**

CO5. Calculate the various parameters of the queuing models. **(PO-a,b)**

DATA COMMUNICATIONS

Course Code : **IS432** **Credits** : **4:0:0**

Prerequisites: **Nil** **Contact Hours : 56**

Course coordinator(s): Suresh Kumar K R

Course objectives:

1. Introduce the students to the concept of Protocol Stacks (OSI & TCP/IP) and basic concepts related to data communication.
2. Create awareness about the different techniques to digital data transmission.
3. Familiarize the students to different strategies for error detection and correction.
4. Make the students appreciate the need for multiple access and different techniques to achieve the same.
5. Give conceptual understanding of different types of networks and connecting devices.

Course Contents:

UNIT -1

Introduction*:* Data communications, Networks, The Internet, Protocols & standards, **Network Models:** Layered tasks, The OSI model, Layers in the OSI model, TCP/IP Protocol suite, Addressing.

UNIT -2

Data & Signals*:* Analog & Digital, Periodic Analog signals, Digital signals, Transmission impairments, Data rate limits, Performance, **Digital Transmission:** Digital-to-Digital conversion - Line coding, Line coding schemes (unipolar, polar, bipolar), Analog-to-Digital conversion: PCM, **Analog Transmission:** Analog to Analog Conversion.

UNIT -3

Error detection & correction: Introduction, Block coding, Linear Block codes, Cyclic codes – CRC, Polynomials, Checksum, **Data link control:** Framing, Flow & error control, Protocols, Noiseless channels (Simplest Protocol, Stop-and-wait protocol).

UNIT -4

Noisy channels (Stop-and-wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ, Piggybacking), **Multiple Access:** Random Access (Aloha, CSMA, CSMA/CD, CSMA/CA), Controlled Access (Reservation, Polling, Token Passing), Channelization (FDMA, TDMA, CDMA).

UNIT -5

Wired LANs*:* IEEE standards, Standard Ethernet, **Wireless LANs:** IEEE 802.11 - Architecture, MAC sublayer, Addressing mechanism, **Connecting LANs, Backbone Networks, & Virtual LANs:** Connecting devices, Backbone networks, Virtual LAN (Intro, Membership, and Configuration).

Text Books:

3. Behrouz A. Forouzan, Data Communications and Networking, Fourth Edition, Tata McGraw-Hill, 2006.

Reference Books:

1. Alberto Leon-Garcia and Indra Widjaja, Communication Networks –Fundamental Concepts and Key architectures, Second Edition, Tata McGraw-Hill, 2004.

2. Wayne Tomasi, Introduction to Data Communications and Networking, Pearson Education, 2005.

Course Outcomes:

The students will be able to-

1. Distinguish different communication models / protocol stacks (OSI & TCP/IP) and solve problems on data transmission by measuring the performance parameters. **(Po-a)**
2. Handle the problems associated with digital data and signals. **(PO-a,b)**
3. Apply different error detection & correction strategies to solve errors induced during data communication. **(PO-b,c)**
4. Use the different strategies of multiple access and achieve better network efficiency. **(PO-b,d)**
5. Illustrate the IEEE standards for different network types and their connecting devices. (**PO-a,e)**

SOFTWARE ENGINEERING

Course Code : **IS433** **Credits** : **3:0:0**

Prerequisites: **NIL** **Contact Hours** : **42L**

Course coordinator(s): Rajaram M Gowda

Course objectives:

- Introduce concept of software engineering and understand SDLC processes.
- Identify the software requirements.
- Explore various design concepts.
- Identify Software Quality Assurance and testing concepts.
- Elucidate Project Management concepts and metrics.

Course Contents:

Unit – 1

Introduction: FAQs about software engineering, Professional and ethical responsibility, Software processes: Software process models, process iteration, process activities, the Rational Unified process, Computer Aided Software Engineering (CASE), Rapid software development: Agile methods, Extreme programming, Rapid application development and software prototyping.

Unit – 2

Software Requirements: Functional and Non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. Requirements Engineering Processes: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. Case studies: ATM, LIBSYS. Object oriented analysis.

Unit – 3

Design Concepts: Design within the context of software engineering, the design process, design concepts, the design model. Architectural Design: Software architecture, Architectural genres, A brief taxonomy of Architectural styles. Component-level design: what is a component? Designing Class-based components. User Interface design: The golden rules, User interface analysis and design, design issues, Static and Dynamic modeling.

Unit – 4

Software Quality Assurance: Background issues, Elements of software quality assurance, SQA tasks, goals, and metrics, Formal approaches to SQA, Statistical software quality assurance. Software Testing Strategies: A strategic approach to software testing, strategic issues, test strategies for conventional software, validation testing, system testing, the art of debugging, Object oriented testing.

Unit – 5

Project management concepts: The management spectrum, people, the product, the process. Process and Project metrics: Metrics in the process and project domains, software measurement, metrics for software quality, integrating metrics within the software process, metrics for small organizations, establishing a software metrics program, CMM, CMMI, PCMM.

Text Books:

1. Roger S. Pressman, Software Engineering-A Practitioners approach, Seventh Edition, McGraw-Hill, 2007.
2. Ian Sommerville, Software Engineering, Eighth Edition, Pearson Education, 2007.

Reference Books:

1. Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering Theory and Practice, Third Edition, Pearson Education, 2006.
2. Waman S Jawadekar, Software Engineering Principles and Practice, Tata McGraw Hill, 2004.
3. Douglas Bell, Software Engineering for Students, A Programming Approach, 4th Edition, Pearson Education.

Course outcomes:

The students will be able to

CO1: Describe software development life cycle processes. **(PO-a,b,h)**
CO2: Analyze software requirements and generate SRS. **(PO-a,b,c,e,i,j)**
CO3: Describe design concepts and develop design document. **(PO-a,b,e,i,j)**
CO4: Describe SQA tasks, goals, and metrics, and test strategies. **(PO-a,b,e)**
CO5: Explain Project management concepts and metrics. **(PO-a,j,k)**

DESIGN AND ANAYSIS OF ALGORITHMS

Course Code : **IS414** **Credits** : **4:0:0**

Prerequisites: **Data Structures** **Contact Hours** : **56**

Course coordinator(s): Dr. Megha. P. Arakeri

Course objectives:

- Introduce the concepts of algorithm and its analysis with respect to time and space.
- Explore the various algorithm design techniques.
- Analyse the time efficiencies of various algorithms.
- Represent the algorithmic time efficiency using different asymptotic notations.
- Identify the limitations of algorithms power.
- Analyse, Design and develop the algorithms to solve problems using the appropriate design technique.

Course Contents:

Unit-I

Introduction: Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types. Fundamentals of the Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical analysis of Non-Recursive and Recursive algorithms.

Unit-II

Brute Force: Selection Sort and Bubble Sort, Divide and Conquer: Merge Sort, Quick Sort, Analysis of Binary Search and Binary Tree Traversal Algorithms, Space and Time Trade-offs: Input Enhancement in String Matching: Horspool's Algorithm

Unit-III

Dynamic Programming: Warshall's and Floyd's Algorithms, The Knapsack Problem and Memory Functions. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees

Unit-IV

Decrease and Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Decrease-by-a-Constant-Factor Algorithms: Fake-Coin Problem, Josephus Problem, Variable-Size-Decrease Algorithms: The Game of Nim. Transform and Conquer: Balanced Search Trees.

Unit-V

Transform and Conquer: Heaps and Heapsort. Limitations of Algorithm Power: P, NP and NP-Complete Problems. Coping with the Limitations of Algorithm Power: Backtracking (n-Queens Problem), Branch-and-Bound (Travelling Salesman Problem), Approximation Algorithms for NP-hard Problems.

Text Book:

1. Anany Levitin ,"Introduction to The Design & Analysis of Algorithms" , 2nd Edition , Pearson Education,2007.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein Introduction to Algorithms,2ndEdition, PHI, 2006.
2. Computer Algorithms, Horowitz E. Sahni S, Rajasekaran S, Galgotia Publications,2001

Course Outcomes:

Students will be able to

CO1. Design an algorithm to solve computational problems and determine its efficiency through mathematical analysis. **(PO-a,b,c)**

CO2. Apply algorithms to solve various computational problems. **(PO-a, f)**

CO3. Classify the problem into P,NP, NP-complete problem types and solve combinatorial optimization problems approximately. **(PO-a, b, f)**

DESIGN AND ANAYSIS OF ALGORITHMS LAB

Course Code : **IS414L** **Credits** : **0:0:1**

Prerequisites: **Data Structures** **Contact Hours** : **28**

Course coordinator(s): Prathima M N

Course Objectives:

- Introduce the students to Brute Force algorithm design techniques and make them realize the weakness of this design technique.
- Familiarize the students with Divide & Conquer, Decrease and Conquer as well as Transform and Conquer Design Techniques and analyze the time efficiencies of these algorithms.
- Acquaint the students with Greedy and Dynamic Programming Design Techniques as well as the concept of back-tracking.
- Enhance written and oral communication skills among students.
- Make the students imbibe the art of writing elegant and efficient programs as well as debugging skills.

Course Contents:

IMPLEMENT THE FOLLOWING USING C/C++ LANGUAGE:

1. Sort a given set of elements using Bubble Sort/Selection Sort and determine the time required to sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm.
2. Sort a given set of elements using Merge Sort method and determine the time required to sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm.
3. Sort a given set of elements using Quick Sort method and determine the time required sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm.
4. Print all the nodes reachable from a given starting node in a digraph using BFS. Give the trace of this algorithm.
5. Sort a given set of elements using the Heap Sort method and determine the time required to sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm.
6. Implement Horspool algorithm for String Matching. Give the trace of this algorithm.
7. Compute the transitive closure of a given directed graph using Warshall's algorithm. Give the trace of this algorithm.
8. Implement Floyd’s algorithm for the All-Pairs- Shortest-Paths problem. Give

the trace of this algorithm.

9. Implement 0/1 Knapsack problem using dynamic programming. Give the trace of this algorithm.
10. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's/Kruskal's algorithm. Give the trace of this algorithm.
11. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. Give the trace of this algorithm.
12. Implement N-Queen's problem using Back Tracking. Give the trace of this algorithm.

Course Outcomes:

Students will be able to

CO1. Identify the need for algorithm design techniques. **(PO-a,b)**

CO2. Implement the algorithms based on various design techniques. **(PO-a,b,d)**

CO3. Analyze the efficiency of various design techniques. **(PO-a,b,d)**

CO4. Produce substantial written documentation. **(PO-j)**

MICROPROCESSORS

Course Code: IS435 **Credits: 4:0:0**
Prerequisites: Nil **Contact Hours: 56L**
Course coordinator(s): Mohan Kumar.S.

Course Objectives

- Learn how the software and hardware and components of a microprocessor-based system work
- Learn both hardware and software devices (such as memory and I/O interfaces) into microprocessor-based system.
- Introduce the concepts of ARM processor Architecture.

Course Contents:

UNIT 1

The Architecture of 8086- Internal Block Diagram of 8086, The Execution Unit, Bus Interface Unit, Addressing Modes,Programming Concepts 1 – The Assembly Process, Assembly for x86, Memory Models, Instruction design.

UNIT 2

Programming Concepts 2 – Approaches to Programming, Data Transfer Instructions, Branch instructions, Arithmetic Instructions, Logical Instructions, Shift and Rotate Instructions, Programming Concepts 3 – String Instructions, Procedures and Macros, Number Format Conversions, ASCII operations, Conversions for computations and Display/Entry, Signed Number Arithmetic, Programming using High Level Language Constructs.

UNIT 3

Programming Concepts 4 – Input/Output Programming, I/O Instructions, Modular Programming, Programming in C with Assembly Modules, The Hardware Structure of 8086 – Pin configuration, Clock, Other Processor Activities, maximum Mode, Instruction Cycle including Delay Loops

UNIT 4

Memory and I/O Decoding – Memory Device Pins, Memory Address Decoding, Memory Banks, I/O Address Decoding, The Interrupt Structure of 8086 – Interrupts of 8086, Dedicated Interrupts of 8086, Software and Hardware Interrupts, Priority of Interrupts, BIOS 10H interrupts, Peripheral Interfacing – Programmable Peripheral Interface – 8255A, modes of operation, Mode 0, Mode 1, Mode 2, Interfacing Digital to Analog Converter, Stepper Motor, Logical Interface, Keypad, Seven Segment Display and Elevator.

UNIT 5

Introduction to Embedded Systems - The RISC Design Philosophy, ARM Design Philosophy, Embedded System Hardware, Embedded System Software, ARM Processor Fundamentals – Registers, Current Program Status Register, Pipeline, Exceptions,

Interrupts, and the Vector Table, Core Extensions, Architecture Revisions, Logical and Compare instruction set and Simple ARM Programs. Development Process in Embedded Systems, The world of ARM.

Text Books

1. Lyla B. Das, "The x86 Microprocessors – Architecture, Programming and Interfacing", Pearson Education, 2013
2. Andrew Sloss, Dominic Symes, Chris Wright, "ARM System Developer's Guide - Designing and Optimizing System Software", Elsevier Publication 2012.
3. Rob Toulson and Tim Wilmshurst "Fast and Effective embedded Systems Design" Elsevier publication 2012.

Reference Books

1. Douglas V.Hall, "Microprocessors and Interfacing Programming and Hardware", Tata McGraw Hill, 2003.
2. Barry B. Brey, "The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386 and 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III Pentium 4 – Architecture,Programming and Interfacing, 7th Edition, PHI, 2008

Course outcomes

Students will be ability to

CO1: Define x86 Arch, memory organization and Assembly instruction. (**POa,POb,POc**)
CO2: Describe the Assembly Language based programming technique. (**POa,POb**)
CO3:Highlighting the Design and development of modular programming.(**POa,POc,POe**)
CO4: Illustrate the interfacing using Programmable Peripheral Interface (**POc,POd**)
CO5: Classifying the ARM architecture and ability to engage in self-learning. (**POc, POl**)

MICROPROCESSORS LAB

Course Code: IS435L **Credits: 0:0:1**
Prerequisites: **Contact Hours: 14P**
Course coordinator(s): Mohan Kumar S

Course Objectives

- Familiarize the architecture of 8086 processor, assembling language programming. Software programming using MASM tool or TASM tool.
- Implement 8086 interfacing with various modules.
- Design of ARM embedded systems, industrial and real time applications by knowing the concepts of microprocessor and embedded systems.

Part A

1a. Write an ALP to search for a key in an array of bytes/words using Binary Search technique.
1b. Write an ALP to count and display number of 1's in the input given using switches of a logical controller interface.

2a. Write an ALP to sort the given array of bytes/words using Bubble Sort technique.
2b. Write an ALP to implement i) Ring Counter ii) BCD up-down Counter using logic controller interface.

3a. Write an ALP to check if the input string matches the password stored earlier. After three attempts the system should stop responding to any further transactions.
3b. Write an ALP to rotate the stepper motor clockwise/anticlockwise or in both directions by x degrees [x will be specified at the time of experiment].

4a. Write an ALP to reverse a given string and check if it is palindrome with/without using string instructions.
4b. Write an ALP to generate half rectified/full rectified wave using DAC interface.

5a. Write an ALP to find the factorial of a number using recursive procedure.
5b. Write an ALP to generate sine wave using DAC interface.

6a. Write an ALP to display 'X' at the center of the screen with the specific attributes [Attributes will be specified at the time of experiment].
6b. Write an ALP to store row number, column number and scan code of key pressed in Keypad interface.

7a. Write an ALP to generate first n Fibonacci numbers.
7b. Write an ALP to implement a simple calculator for add/subtract operations using Keypad interface.

8a. Write an ALP to display system time.
8b. Write an ALP to display FIRE and HELP alternatively n number of times on seven segment display interface.

9a. Write an ALP to display decimal counter to count from 00-99 at the center of the screen.
9b. Write an ALP to display n character message on seven segment display in a rolling fashion.

10a. Write an ALP to open, read and close a file. Display the contents of the file on screen.
10b. Write an ALP to move the elevator to the first request floor and bring it back to ground floor. The elevator need not respond to the intermediate requests in both directions.

Part B using ARMSIm1.9.1 Tool

1. Write a ARM program to Add two 16 bit numbers
2. Write a ARM program to search for a key in an array using any search technique.
3. Write a ARM program to display character from 0-9
4. Design and develop basic calculator using ARMSim191 Tool

Course Outcomes

Students will be able to
C01: Design and Develop Assembly level programming and apply working of 8086.

(Po-c,d)

CO2: Compare the various interface techniques. Design and develop the programs for various interfacing models**(PO-b,c,d)**

C03: Learning the Communication Standards ARM embedded programming using ARMSim Tool and Cortex Board. **(PO-a,c,d,l)**

Finite Automata and Formal Languages

Course Code : **IS416** **Credits** : **3:1:0**

Prerequisites: **Nil** **Contact Hours** : **42L+14T**

Course coordinator(s): Deepthi. K

Course objectives:

- Define the basic concepts of Automata Theory, the need and design for computation models.
- Write the Regular Expression (RE) for the given FA and prove their properties.
- Design CFG and their variants.
- Design the PDA and DPDA for the given language or the CFG.
- Design the TM and discuss their variants.

Course Contents:

Unit 1

Finite Automata and Regular Expressions: Introduction to Finite Automata: The central concepts of Automata theory, Deterministic finite automata, Nondeterministic finite automata, An application of finite automata, Finite automata with Epsilon-transitions

Unit 2

Regular Languages, Properties of Regular Languages: Regular expressions; Finite Automata and Regular Expressions, Regular languages: Proving languages not to be regular languages, Closure properties of regular languages, Equivalence and minimization of automata

Unit 3

Context-Free Grammars and Languages: Context free grammars, Parse trees: Constructing parse trees, The yield of a parse tree, Applications, Ambiguity in grammars and Languages, Normal forms for CFGs; The pumping lemma for CFGs

Unit 4

Pushdown Automata and Properties of Context-Free Languages: Closure properties of CFLs, Definition of the Pushdown automata: The languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.

Unit 5

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

Tutorial Contents:

1. The design of DFA.
2. Converting NFA to DFA.
3. Converting Ɛ-NFA to DFA.
4. Constructing Regular expression given a regular language.
5. Proving languages not to be regular using pumping lemma for regular languages.
6. Minimization of DFA.
7. Constructing Context free grammar given a context free language.
8. Simplification of CFGs
9. Designing Deterministic and Non Deterministic PushDown Automata.
10. Converting PDA to CFG and Viceversa.
11. Design of Turing Machine given Recursively enumerable language.

Text Book

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata,Theory, Languages and Computation, 3rd Edition, Pearson education, 2007

Reference Books

1. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.
2. Daniel I.A. Cohen: Introduction to Computer Theory, 2nd Edition, John Wiley & Sons, 2004.