M.S. RAMAIAH INSTITUTE OF TECHNOLOGY

BANGALORE

(Autonomous Institute, Affiliated to VTU)

SYLLABUS

(For the Academic year 2016 – 2017)

III & IV Semester B.E

Information Science and Engineering

History of the Institute:

M. S. Ramaiah Institute of Technology was started in 1962 by the late Dr. M.S. Ramaiah, our Founder Chairman who was a renowned visionary, philanthropist, and a pioneer in creating several landmark infrastructure projects in India. Noticing the shortage of talented engineering professionals required to build a modern India, Dr. M.S. Ramaiah envisioned MSRIT as an institute of excellence imparting quality and affordable education. Part of Gokula Education Foundation, MSRIT has grown over the years with significant contributions from various professionals in different capacities, ably led by Dr. M.S. Ramaiah himself, whose personal commitment has seen the institution through its formative years. Today, MSRIT stands tall as one of India's finest names in Engineering Education and has produced around 35,000 engineering professionals who occupy responsible positions across the globe.

About the Department:

Department of Information Science and Engineering (ISE) was established in the year 1992 with an objective of producing high-quality professionals to meet the demands of the emerging field of Information Science and Engineering. Department started with UG programme with an annual sanctioned intake of 30 students and the intake was enhanced to 60 seats in the year 1999, to 90 seats in the year 2001 and then to 120 seats in the year 2008. Department started M.Tech in Software Engineering in the year 2004. Department is recognized as R&D center by VTU in the year 2012. Department has well equipped laboratories. Some of the laboratories have also been set up in collaboration with industries such as Intel, Apple, Honeywell, EMC^2, Microsoft and IBM. Department has highly qualified and motivated faculty members. All faculty are involved in research and technical paper publications in reputed technical journals, conferences across the world. Department has been accredited by the NBA in 2001, 2004, 2010 and reaccredited in the year 2015 based on Outcome Based Education format. The department is successfully conducting conferences & workshops for students and academicians in the emerging technologies.

Faculty List

Sl.No.	Name of Faculty	Qualification	Designation
1	Dr. Vijaya Kumar B P	M.Tech (IITR), Ph.D (IISc.)	Professor & Head
2	Dr. Lingaraju G M	Ph.D	Professor
3	Dr.Mydhili K Nair	M.Tech, Ph.D (Anna University)	Professor
4	Rajaram M Gowda	MTech. (Ph.D)	Associate Professor
5	Shashidhara H S	M.Tech. (Ph.D)	Associate Professor
6	George Philip C	M.Tech.	Associate Professor
7	Dr. Megha.P.Arakeri	M.Tech, Ph.D (NITK)	Associate Professor
8	Dr. Siddesh G M	M.Tech., Ph.D	Associate Professor
9	T Tamilarasi	M.E (Ph.D)	Assistant Professor
10	Savita K Shetty	M.Tech.	Assistant Professor
11	Myna A N	M.Sc(Engg) , (Ph.D)	Assistant Professor
12	Deepthi K	M.Tech.	Assistant Professor
13	Lincy Meera Mathews	M.Tech (Ph.D)	Assistant Professor
14	Dr. P M Krishna Raj	M.Sc(Engg), Ph.D	Assistant Professor
15	Rajeshwari S B	M.Tech.	Assistant Professor
16	Prathima M N	M.E.	Assistant Professor
17	Pushpalatha M N	M.Tech (Ph.D)	Assistant Professor
18	Mohan Kumar S	M.Tech (Ph.D)	Assistant Professor
19	Sumana M	M.Tech (Ph.D)	Assistant Professor
20	Prashanth Kambli	M.Sc, (MSc(Engg.) by Research)	Assistant Professor
21	Naresh E	M.Tech, (Ph.D)	Assistant Professor
22	Jagadeesh Sai D	M.Tech.	Assistant Professor
23	Mani Sekhar S R	M.Tech.(Ph.D)	Assistant Professor
24	Suresh Kumar K R	M.Tech.(Ph.D)	Assistant Professor
25	Sunitha R S	M.Tech.	Assistant Professor
26	Sandeep B L	M.Tech.(Ph.D)	Assistant Professor
27	Dr. Dayananda P	M.Tech, Ph.D	Assistant Professor
28	Koushik S	M.Tech.(Ph.D)	Assistant Professor

Vision and Mission of the Institute and the Department

Vision of MSRIT

To evolve into an autonomous institution of international standing for imparting quality technical education

Mission of MSRIT

MSRIT shall deliver global quality technical education by nurturing a conducive learning environment for a better tomorrow through continuous improvement and customization

Quality Policy

"We at M. S. Ramaiah Institute of Technology, Bangalore strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management system Complemented by the Synergistic interaction of the stake holders concerned".

Vision of Department

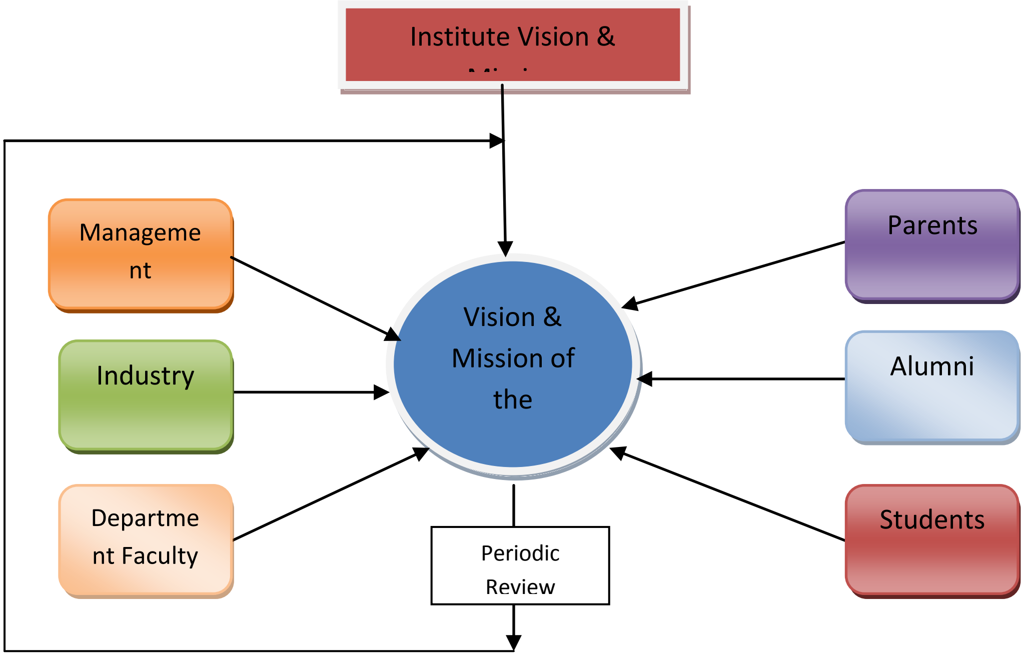
To evolve as an outstanding education and research center of Information Technology to create high quality Engineering Professionals for the betterment of Society

Mission of Department

- To provide a conducive environment that offers well balanced Information Technology education and research
- To provide training and practical experience in fundamentals and emerging technologies
- To nurture creativity for overall personality development

Process of deriving the vision and mission of the department

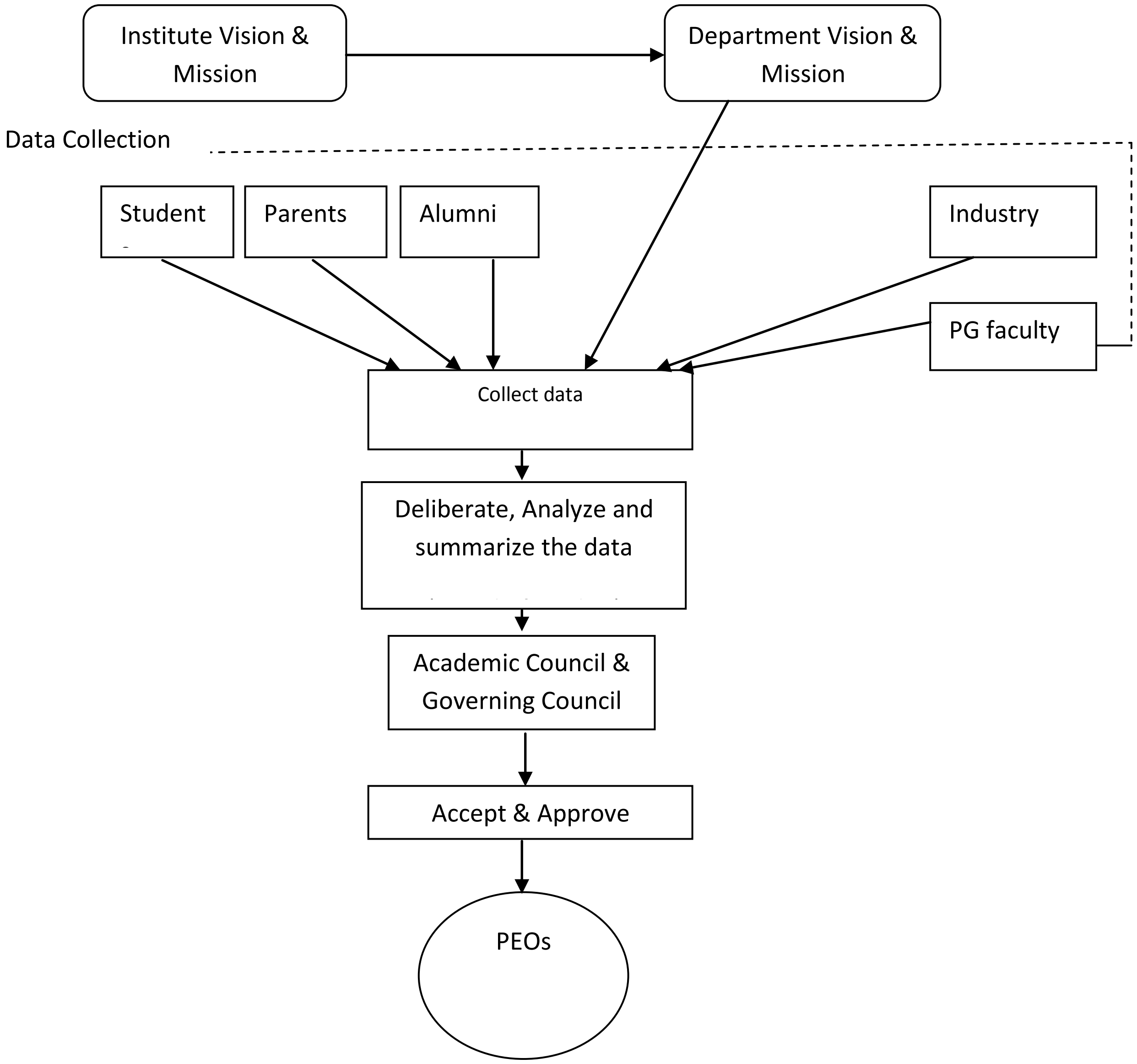
Process of deriving the vision and mission of the department is shown in Figure below:



Programme Educational Objectives (PEOs) of the programme

PEO1: Become competent Information Technology professionals with continuous progress in career or learning
PEO2: Design and develop computing systems using modern tools and technologies
PEO3: Function effectively as professionals in a team environment

Process of Deriving Programme Educational Objectives


Institute Vision & Mission
Department Vision & Mission
Data Collection
Student
Parents
Alumni
Industry
PG faculty
Collect data
Deliberate, Analyze and summarize the data
Academic Council & Governing Council
Accept & Approve
PEOs

PROGRAM OUTCOMES as defined by NBA in the revised format:

Engineering Graduates will be able to:

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

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **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.

4. **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.

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

6. **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.

7. **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.

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

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

10. **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.

11. **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.

12. **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.

Programme Specific Outcomes (PSOs)

A graduate in ISE program will demonstrate:

PSO1 – **Problem Solving Skills,** ability to understand and analyze the Information Technology problems and develop computer programs

PSO2 –**Applied Engineering Skills**, ability to apply standard practices and strategies in Software Development

PSO3 – **Communication and Entrepreneurship skills,** ability to exchange knowledge and incubate ideas

Mapping of PEO's and PO's

The correlation between the Programme outcomes and Program Educational objectives are mapped in the Table shown below:

Correlation between the POs and the PEOs

PEO	**Programme Outcomes**											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
Become competent Information Technology professionals with continuous progress in career or learning	X	X	X	X	X	X	X	X		X		
Design and develop computing systems using modern tools and technologies			X	X	X	X	X				X	X
Function effectively as professionals in a team environment								X	X	X	X	X

Curriculum breakdown structure:

The curriculum of Information Science and Engineeering programme is so structured to include all the courses that together satisfy the requirements of the programme specific criteria prescribed by the **CSAB**(Comptuter Science Accreditation Board), IEEE Computer Society/Association for Computing Machinery (ACM). The Course code, Course title, the number of contact hours and the number of credits for each course are given in the following table. The courses are grouped in line with the major components of the curriculum namely: (i) Mathematics and Basic sciences, (ii) Basic Engineering courses, (iii) Humanities and Social Sciences, (iv)Professional core courses, (v) Electives and (vi) Industry exposure/internship.

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

Board of Studies

S. No.	Category	Name of the Person with Official Address	Status
1	Head of the Department Concerned	Dr. Vijaya Kumar B P Head of the Department Information Science & Engg., M.S. Ramaiah Institute of Technology, MSRIT Post, Bangalore – 560 054.	Chair Person
2	Faculty members nominated by the Academic Council	Shashidhara H S Associate Professor Dept. of Information Science & Engineering, M.S. Ramaiah Institute of Technology, MSR Post, Bangalore – 560 054.	Member
		Dr. Megha P Arakeri Associate Professor Dept. of Information Science & Engineering, M.S. Ramaiah Institute of Technology, MSR Post, Bangalore – 560 054.	Member
		Dr. Siddesh G M Associate Professor, Dept. of Information Science & Engineering, M.S. Ramaiah Institute of Technology, MSR Post, Bangalore – 560 054.	Member
		Dr. P. M Krishna Raj Asst. Professor Dept. of Information Science & Engineering, M.S. Ramaiah Institute of Technology, MSR Post, Bangalore – 560 054.	Member
3	Experts in the subject from outside the College, to be nominated by the Academic Council	Dr. Satish Babu Professor & Head, Dept. of Computer Science & Engineering SiddagangaInstitute of Technology NH 206,B.H.Road, Tumkur, Karnataka 572103. bsbsit@gmail.com	Autonomous Institute Member
		Dr. Dilip Kumar Professor & Head University Visvesvaraya College of Engineering (UVCE),	Government University Member

		K R Circle, Dr Ambedkar Veedhi, Bangalore Karnataka 560001	
		Dr.Y.N. SRIKANT Professor, Department of Computer Science and Automation, Indian Institute of Science Bangalore 560 012.	VTU Member from IISc
		Madhu N. Belur Department of Electrical Engineering Indian Institute of Technology Bombay Powai, Mumbai 400 076 India	Special Invitee
		Dr. Chetan Kumar S Manager Software Development CISCO System Cessna Business Park, Kadubeesanahalli Village, Varthur Hobli, Sarjapur Marathalli, Bangalore – 560 087	Expert Member from Industry
		Mr.Niranjan Salimath Beaglesloft , 37/5, Ulsoor Rd, Yellappa Chetty Layout, Sivanchetti Gardens, Bengaluru, Karnataka 560042	Alumni Member

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

(Autonomous Institute, Affiliated to VTU)

SCHEME OF TEACHING FOR THE ACADEMIC YEAR 2016-2017

III SEMESTER B.E. INFORMATION SCIENCE AND ENGINEERING

Sl.No	SubjectCode	Subject	Credits*				
			L	T	P	S	Total
1	ISMAT311	Mathematics-III	3	1	0	0	04
2	IS332	Computer Organization and Architecture	4	0	0	0	04
3	IS333	Data Structures	4	0	0	0	04
4	IS333L	Data Structures Lab	0	0	1	0	01
5	IS314	Discrete Mathematical Structures	3	1	0	0	04
6	IS315A/ IS315B	Object Oriented Programming with C++ / Object Oriented Programming with Java	4	0	0	0	04
7	IS315L	Object Oriented Programming Lab	0	0	1	0	01
8	IS346	Management & Entrepreneurship	2	0	0	1	03
		Total	**20**	**2**	**2**	**1**	**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 2016-2017

IV SEMESTER B.E. INFORMATION SCIENCE AND ENGINEERING

Sl.No	SubjectCode	Subject	Credits*				
			L	T	P	S	Total
1	ISMAT411	Mathematics-IV	3	1	0	0	04
2	IS442	Data Communications	3	0	0	1	04
3	IS433	Software Engineering	3	0	0	0	03
4	IS414	Design & Analysis of Algorithms	4	0	0	0	04
5	IS414L	Design & Analysis of Algorithms Lab	0	0	1	0	01
6	IS435	Microprocessors	4	0	0	0	04
7	IS435L	Microprocessors Lab	0	0	1	0	01
8	Is416	Finite Automata & Formal Languages	3	1	0	0	04
		Total	**20**	**2**	**2**	**1**	**25**

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

ENGINEERING MATHEMATICS - III

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

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 1, 2)(PSO 2)**

CO2. Fit a suitable curve by the method of least squares and determine the lines of regression fora set of statistical data. **(PO 1, 2) (PSO 2)**

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 1, 2) (PSO 2)**

CO4. Find Fourier transforms Fourier sine and Fourier cosine transforms of functions and solving difference equations using Z-transforms. **(PO 1, 2) (PSO 2)**

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 1, 2) (PSO 2)**

Computer Organization and Architecture

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

Prerequisites: NIL **Contanct Hours: 56**

Course coordinator(s): George Philip C

Course objectives:

- Introduction to combinational circuits.
- Introduction to sequential Circuits
- Study of Computer Organization, structure and functions through instruction execution, memory and I/O
- Study of internal structure and functioning of CPU including instruction execution
- Introduction to modern computer architectures

Course Contents:

Unit–I

Combinational Logic: Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers.

Unit–II

Sequential Logic: Flip-Flops.

Registers and Counters: Registers, Shift Registers, Ripple Counters, Synchronous Counters, Other Counters.

Unit–III

Overview, the Computer System: Organization and Architecture, Structure and Function, Performance Assessment-Clock Speed and IPS, Benchmarks. Computer Components, Instruction Fetch and Execute. Interconnection Structures, Bus Interconnection. Computer Memory System Overview, Cache Memory Principles, Programmed IO, Interrupt Processing, Direct Memory Access Function.

Unit–IV

The CPU: The Arithmetic and Logic Unit (ALU), Addressing Modes, Instruction Formats, Assembly Language, Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining Strategy, Pipeline Hazards, Dealing with Branches.

Unit–V

Parallel Organization: Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and the MESI protocol, Multithreading and Chip Multiprocessors, Clusters, Non Uniform Memory Access, 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, 9e, 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.

Course outcomes:

Students will be able to

CO1. Design basic combinational circuits. **(PO 1, 2, 4) (PSO 2)**

CO2. Design basic sequential circuits.**(PO- 1, 2, 4) (PSO 2)**

CO3. Describe the structure and functions of computer components, interconnection structures, memory and I/O **(PO 1, 7) (PSO 2, 3)**

CO4. Describe the internal structure and functions of CPU including instruction execution. **(PO- 1, 7) (PSO 2 , 3)**

CO5. Describe modern computer architectures (**PO 1, 12) (PSO 2, 3)**

DATA STRUCTURES

Course Code : IS333 **Credits : 3:0:0**

Prerequisites:CS101/CS201 **Contact Hours : 42**

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 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; Recursion: Recursive definition and processes, Examples – Factorial, Tower of Hanoi;

Unit -II

The Queues: Definition, Representation, Primitive operations of queue and its implementation; Circular queues and Priority queues;

Unit -III

The Linked List: Memory allocation functions; Representation and implementation of operations (Insertion, Deletion and Search) of Singly, Doubly and Circular Linked Lists, Comparing the dynamic and array implementation of lists, Implementation of Header Nodes

Unit-IV

Applications: Implementation of stack and queue using lists; Trees: Binary Trees, Binary Tree Representations, Representing Lists as Binary trees, Trees and their applications; Binary Search Tree

Unit- V

Introduction to Files, Introduction to File Structures, Comparing Data Structures with FileStructures, Short history of file structure design, Differentiating between Physical files and logical files,B-Tree: Searching, Insertion and Deletion; B+ Tree: Searching, Insertion and Deletion; Hashing, Hash Function, Collision, Probability of Collision, Collision handling techniques, Progressive Overflow, Buckets, Chained Progressive Overflow.

Text Books

1. Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C", 2nd Edition, PHI, 2009.
2. Michael J. Folk, Bill Zoellick and Greg Riccardi, "File Structures-An Object Oriented Approach with C++", Pearson Education, 2004

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. Implement stack data structures and its applications **(PO 1, 2, 3) (PSO 1, 2)**

CO2. Explain the working principle of various types of queues, their applications and implementations **(PO1, 2, 3) (PSO 1, 2, 3)**

CO3. Develop and implement various types of linked lists using dynamic memory allocation **(PO1, 2, 3) (PSO 1, 2)**

CO4. Describe the concept of non linear structures, their Design and their implementation **(PO 1, 2, 3) (PSO 1, 2, 3)**

CO5. Understand the concept of B-trees, B+trees and their operations **(PO 1, 2, 3) (PSO 1, 2)**

Data Structures Lab

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

Prerequisites:**cs101/201** **Contact Hours: 14**

Course coordinator(s): Deepthi. K

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:

Part - A

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.

Part – B

Some of the Mini Project Topics:

1. Huffmann Coding
2. Banking system - Sorting and Searching problem
3. Dijkstras Algorithm
4. Hashing and Hash table generation
5. Breadth First Search and Depth First Search Implementation
6. B-tree implementation

Text Book:

3. Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C".
4. 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:
Student will be able to:

CO1. Design and implement the concepts of data structures **(PO 1, 2, 3, 4) (PSO 1, 2)**
CO2. Appreciate the usage of data structures in various domains **(PO 1, 2) (PSO 1, 2)**
CO3. Produce a substantial written documentation **(PO 10) (PSO 3)**

DISCRETE MATHEMATICAL STRUCTURES

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

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 ComputerScience , 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-1,2) (PSO 2)**

CO2: Apply the properties of relations on real world examples. **(PO-1,2) (PSO 2)**

CO3: Develop structures useful in set theory, algebra by the study of partially ordered sets and lattices. **(PO-1,2) (PSO 1, 2)**

CO4: Apply the concepts of graphs in computers and several other applications. **(PO-1,3) (PSO 2)**

CO5: Analyze the different mathematical structures and techniques by introducing computing applications. **(PO-1,2) (PSO 1)**

OBJECT ORIENTED PROGRAMMING USING C++

Course Code : **IS315A** **Credits** : **4:0:0:0**

Prerequisites:**NIL** **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 1, 2, 12) (PSO 1)**

CO2: Understanding the mechanism of deriving a new class from older classes through inheritance.**(PO 1, 3, 5) (PSO 1, 2)**

CO3: Constructing applications to provide flexible options for the creation of new definitions for some of the operators.**(PO 1, 2, 3, 5) (PSO 2)**

CO4: Building of programs for automatic initialization of objects and destroy objects that are no longer required.**(PO 1, 2, 3, 5) (PSO 1)**

CO5. Designing a program using Templates & Exception Handling. **(PO 1, 2, 5, 12) (PSO 1, 2)**

OBJECT ORIENTED PROGRAMMING USING JAVA

Course Code : **IS315B** **Credits** : **4:0:0:0**

Prerequisites:**NIL** **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 Java.
- Specifying mechanism of deriving a new class from older classes through inheritance.
- Implement methods to select appropriate member function during run time.
- Design a program using standard libraries and Exception Handling

UNIT – 1

The Java Language: The History and Evolution of Java, Java's Lineage, Java Applets, Security, Portability,Java's Magic: The Bytecode, Servlets: Java on the Server Side, The Java Buzzwords

Object-Oriented Programming, Two Paradigms, Abstraction, The Three OOP Principles

UNIT – 2

Java Language - Data Types, Variables, and Arrays, Operators

Control Statements, Java's Selection Statements, if, switch, Iteration Statements, while, do-while, for, The For-Each Version of the for Loop, Nested Loops, Jump Statements, Using break, Using continue

UNIT - 3

Introducing Classes, Class Fundamentals, The General Form of a Class, A Simple Class, Declaring Objects, A Closer Look at new, Assigning Object Reference Variables, Introducing Methods, Adding a Method to the Box Class, Returning a Value, Adding a Method That Takes Parameters, Constructors, Parameterized Constructors, The this Keyword, Instance Variable Hiding, Garbage Collection, The finalize( ) Method, A Stack Class

A Closer Look at Methods and Classes, Overloading Methods, Overloading Constructors, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final

UNIT - 4

Arrays Revisited, Introducing Nested and Inner Classes, Exploring the String Class, Using Command-Line Arguments, Varargs: Variable-Length Arguments, Overloading Vararg Methods, Varargs and Ambiguity

Inheritance - Inheritance Basics, Member Access and Inheritance, A More Practical Example, A Superclass Variable Can Reference a Subclass Object, Using super, Using super to Call Superclass, Constructors, A Second Use for super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Why Overridden Methods?, Applying Method Overriding, Using Abstract Classes, Using final with Inheritance, Using final to Prevent Overriding, Using final to Prevent Inheritance, The Object Class

UNIT - 5

Packages and Interfaces, Packages, Defining a Package, Finding Packages and CLASSPATHA Short Package Example, Access Protection, An Access Example, Importing Packages, Interfaces, Defining an Interface, Implementing Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces, Interfaces Can Be Extended, Default Interface Methods, Default Method Fundamentals, A More Practical Example, Multiple Inheritance Issues, Use static Methods in an Interface, Final Thoughts on Packages and Interfaces Exception Handling, Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Displaying a Description of an Exception, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java’s Built-in exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Three Recently Added Exception FeaturesUsing Exceptions

Text Books:

Herbert Schildt, Java: The Complete Reference, 9th Edition, McGraw Hill

Course Outcomes

Student will be able to:

CO1: Describe fundamental object oriented principles **(PO 1, 2, 12) (PSO 1)**

CO2: Write programs using java syntax and semantics.**(PO 1, 3, 5) (PSO 2)**

CO3: Create classes for a given problem description**(PO 1, 2, 3, 5) (PSO 1, 2)**

CO4: Write reusable modules using inheritance (**PO 1, 2, 3, 5) (PSO 1, 2)**

CO5: Create packages to group related modules.**(PO 1, 2, 5, 12) (PSO 2)**

OBJECT ORIENTED PROGRAMMING LAB

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

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) Write a C++ program to illustrate the concept of function overloading, inline, friend function and default arguments
2) Write a C++ program to demonstrate the use of friend class and array of structure
3) Write a C++ program to demonstrate the concept of classes and constructors.
4) 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.
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 overload binary, relational and subscript operators, Display the results by overloading the operator <<.
8) Write a C++ program to Implement the operations by overloading the Unary, Assignment, new operators and display the results by <<.
9) Write a C++ Program to implement Type Conversion
10) Write a C++ Program to implement virtual functions and dynamic polymorphism
11) Write a C++ program to illustrate the File handling
12) Write C++ program to use try catch statements to handle some predefined Exceptions and Used Defined Exceptions
13) Write a C++ program to illustrate the Function templates and class templates.
14) Write a C++ Program to Illustrate the concept of STL

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 1, 2, 11) (PSO 1, 2)**

CO2. Building of programs for automatic initialization of objects and destroy objects that are no longer required.**(PO 1, 5, 7, 11) (PSO 1, 2)**

CO3. Specifying mechanism of deriving a new class from older classes through Inheritance. **(PO 2, 3, 5) (PSO 2)**

CO4. Constructing applications to provide flexible options for the creation of new definitions for some of the operators. **(PO 1, 2 11) (PSO 1, 2)**

CO5. Designing a program using Templates & Exception Handling.**(PO 1, 10, 11) (PSO 1, 2)**

MANAGEMENT AND ENTREPRENEURSHIP

Course Code : IS346	**Credits : 2:0:0:1**
Prerequisites : NIL	**Contact Hours : 28L**

Course coordinator(s): Suresh Kumar L

Course Objectives

1. Provide an insight into management and various approaches used
2. Describe the various functionalities of management
3. Provide the knowledge and skills required to become an entrepreneur
4. Discuss the process of setting up a small business
5. Lifelong learning through self study

UNIT-I

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

UNIT II

PLANNING: Nature, Importance, Types of plans (Definitions and Meaning only), Steps, DECISION MAKING: Meaning, Types, Steps, COORDINATION: Need for Coordination, Requisites for Excellent Coordination, Types of Coordination, DIRECTION AND SUPERVISION: Giving Orders, Motivation- Meaning, Nature, Motivation Theories-Maslow's Theory, Herzberg's Theory, McClelland's Need for Achievement Theory

UNIT III

COMMUNICATION: Importance, Purposes, Formal Communication, Forms of Communication, Informal communication, 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-IV

ENTREPRENEURSHIP: Importance, Concepts, Characteristics of a Successful Entrepreneur 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.

UNIT-V

Self Study:

Study of

- NASSCOM, TIE
- Government schemes like TBI (technology business Incubation) and NAIN (New Age Incubation Network)
- Eco systems required to start an industry - DIC, MSME, CSIR, KSSIDC, KIADB, KSFC, TECKSOK
- Similarly eco systems available for e-commerce like funding agencies in private sector such as early funding agencies, angel Investors, venture capitalists (ex: http://axilor.com/), New Venture Creation (www.celbits.org/nvc).
- Studying B schools like IIM for their Innovation in management and recent trends in management techniques (journals).

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

CO 1: Analyze the various approaches of management **(PO 8) (PSO 1, 2)**

CO 2: Apply the administrative skills of planning, decision making and coordinating **(PO 8) (PSO 2, 3)**

CO 3: Analyze motivation theories and apply them to direct and supervise employees **(PO 8, 9, 10) (PSO 1, 2, 3)**

CO 4: Apply effective communication and leadership skills **(PO 8, 9, 1 0) (PSO 3)**

CO 5: Identify the skills required to become a successful entrepreneur **(PO 8, 10, 12) (PSO 3)**

ENGINEERING MATHEMATICS - IV

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

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 1, 2) (PSO 2)**

CO2. Find solution of partial differential equations by direct integration method and separation of variables. **(PO 1, 2) (PSO 2)**

CO3. Express the probability distribution arising in the study of engineering problems and their applications. **(PO 1, 2) (PSO 2)**

CO4. Apply the stochastic process and Markov Chain in prediction of future events. **(PO 1, 2) (PSO 2)**

CO5. Calculate the various parameters of the queuing models. **(PO 1, 2) (PSO 2)**

DATA COMMUNICATIONS

Course Code :IS442 **Credits : 3:0:0:1**

Prerequisites: Nil **Contact Hours : 42L**

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 digital data transmission.
3. Familiarize the students to different strategies for error detection and correction.
4. Make the students appreciate the need for multiple accessand different techniques to achieve the same.
5. Give conceptual understanding of different types of networks and connecting devices.

Course Contents:

UNIT -1

Introduction: Datacommunications – Introduction, components, Data Representation, Data Flow; Networks – Network criteria, Physical Structures, Network Models, Categories of networks;Protocols, Standards;**NetworkModels:**Layeredtasks;TheOSImodel – Layered architecture, Peer-to-Peer Process, Encapsulation;LayersintheOSImodel – introduction to and the different services offered by all seven layers; TCP/IPProtocolsuite; Addressing.

Self Study: TheInternet – Brief history, Internet today; Standards organization; OSI model Vs TCP/IP model; Protocol stacks used in technologies like WAP, IOT, GSM and GPRS;

UNIT -2

Data&Signals*:*Analog&Digital – data, signals, periodic and nonperiodic; Transmission of digital signal; TransmissionImpairments – Attenuation, Distortion, Noise; Dataratelimits – Nyquist Bit Rate, Shannon Capacity; Performance, **DigitalTransmission:**Digital-to-Digitalconversion-Linecoding,Linecodingschemes (unipolar, polar, bipolar);Analog-to-Digitalconversion:PCM.

Self Study: Analog signals – Sine wave, phase, wavelength, time and frequency domains, composite signals, bandwidth; Digital signals – Bit rate, Bit Length, Digital

Signal as a composite analog signal; The What, Why and How of Audio and Video Compression (mp3, mp4 etc).

UNIT -3

Errordetection&correction: Introduction,Block coding, Linear Block codes, Cycliccodes – CRC,Polynomials,Checksum, **Datalinkcontrol:**Framing,Flow&errorcontrol,Protocols,Noiselesschannels (Simplest Protocol, Stop-and-wait protocol); Noisychannels (Stop-and-wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ, Piggybacking).

Self Study: HDLC – Configuration and transfer modes, frames, control field; Point-to-Point Protocol – framing, transition phases.

UNIT -4

Multiple Access: Random Access (Aloha, CSMA, CSMA/CD, CSMA/CA), Controlled Access (Reservation, Polling, Token Passing), Channelization (FDMA, TDMA, CDMA)

Self Study: Design and Simulation of sample networks with different network topologies and their performance comparison using NS2

UNIT -5

WiredLANs:IEEEstandards;StandardEthernet;**WirelessLANs:**IEEE802.11-Architecture,MACsublayer,addressingmechanism; Connectingdevices,Backbonenetworks

Self Study: Comparison on different connecting devices; the What, Why and How of Bluetooth & Adhoc Network; the What, Why and How of Zigbee & PAN (Personal Area Networks); the What, Why and How of Virtual LAN.

Text Books:

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

References:

1. Publications and websites (will be specified later)
2. Alberto Leon-Garcia and Indra Widjaja, Communication Networks –Fundamental Concepts and Key architectures, Second Edition, Tata McGraw-Hill, 2004.
3. 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 1, 2, 4) (PSO 1, 2)**
2. Handle the issues associated with digital data signals and solve the problems on data transmission by measuring the performance parameters. **(PO 1, 2, 3, 4) (PSO 1, 2)**
3. Apply different error detection, error correction as well as flow control strategies to solve error and flow control issues induced during data communication. **(PO 2, 3, 4) (PSO 2)**
4. Use the different strategies of multiple access to achieve better network efficiency and analyze the network performance through simulation.**(PO 2, 3, 4) (PSO 1, 2)**
5. Illustrate the IEEE standards for wired, wireless LANs and their connecting devices. **(PO 10) (PSO 3)**

SOFTWAREENGINEERING

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

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 1, 2, 8) (PSO 3)**

CO2: Analyze software requirements and generate SRS.**(PO 1, 2, 3, 5, 9 ,10) (PSO 1)**

CO3: Describe design concepts and develop design document. **(PO-1, 2, 5, 9 , 10) (PSO 2, 3)**

CO4: Describe SQA tasks, goals, and metrics, and test strategies. **(PO 1, 2, 5) (PSO 1, 3)**

CO5: Explain Project management concepts and metrics.**(PO 1, 10, 12) (PSO 1, 3)**

DESIGN AND ANAYSIS OF ALGORITHMS

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

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 thelimitations 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 1, 2, 3) (PSO 1, 2)**

CO2. Apply algorithms to solve various computational problems.**(PO 1, 6) (PSO 2)**

CO3. Classify the problem into P,NP, NP-complete problem types and solve combinatorial optimization problems approximately.**(PO 1, 2, 6) (PSO 1, 2)**

DESIGN AND ANAYSIS OF ALGORITHMSLAB

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

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 Sortand 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’salgorithm. 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 1, 2) (PSO 1)**

CO2. Implement the algorithms based on various design techniques.**(PO 1, 2, 4) (PSO 2)**

CO3. Analyze the efficiency of various design techniques.**(PO 1, 2, 4) (PSO 1, 3)**

CO4. Produce substantial written documentation.**(PO 10) (PSO 3)**

MICROPROCESSORS

Course Code: IS435 **Credits: 4:0:0:0**
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:

UNIT1

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.

UNIT2

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.

UNIT3

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

UNIT4

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.

UNIT5

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.

ReferenceBooks

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

<u>Course outcomes</u>

Students will be able to

CO1: Define x86 Arch, memory organization and Assembly instruction. (**PO 1, 2, 3**) **(PSO 1, 3)**

CO2: Describe the Assembly Language based programming technique. (**PO 1 ,2**) **(PSO 3)**

CO3: Highlighting the Design and development of modular programming.(**PO 1 3 5**) **(PSO 3)**

CO4: Illustrate the interfacing using Programmable Peripheral Interface (**PO 3, 4**) **(PSO 1)**

CO5: Classifying the ARM architecture and ability to engage in self-learning. (**PO 3, 12**) **(PSO 1, 3)**

MICROPROCESSORS LAB

Course Code: IS435L **Credits: 0:0:1:0**
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 ARMSim Tool

1. Write an ARM program to add two numbers of different sizes.
2. Write an ARM program to search for a key in an array using Linear Search method. Display appropriate messages on LCD screen. Make both the LEDs on interface ON if the key is present.
3. Write an ARM program to display 0-9 on seven segment display.
4. Write an ARM program which display up counter (0-9) when left button is pressed and down counter (9-0) when right button is pressed.
5. Demonstrate the basic calculator for addition and subtraction operations using keypad and lcd display interface in ArmSim

Course Outcomes:

Students will be able to
CO1: Write assembly code for 8086 processor **(PO 1, 3, 9) (PSO 2)**
CO2: Interface peripherals to 8086 processor **(PO 1, 3, 9) (PSO 1, 2)**
CO3:Demostrate the knowledge of working with assemblers and debuggers **(PO 5) (PSO 3)**
CO4: Write assembly code using ARM instructions **(PO 1, 3, 9, 12) (PSO 2, 12)**
CO5: Demonstrate the usage of ARMSim tool **(PO 5) (PSO 3)**

Finite Automata and Formal Languages

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

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.

COURSE OUTCOMES

By the end of the course, students will be able to:

CO1: Describe the basic concepts of automata theory and design the Deterministic Finite State Models. **(PO 1) (PSO 1, 2)**

CO2: Formulate Regular Expression (RE) for Finite Automata and vice versa. **(PO 1, 2) (PSO 1, 2)**

CO3: Design Context Free Grammar (CFG) in various Normal Forms and discuss their Applications. **(PO 1, 2) (PSO 1, 2)**

CO4: Design and Convert N-Push Down Automata and D-Push Down Automata for CFG and vice versa. **(PO 1, 2) (PSO 2)**

CO5: Design Turing Machine for a given language and discuss its variants. **(PO 1, 2) (PSO 2)**