RAMAIAH
Institute of Technology

CURRICULUM

for the Academic year 2021 – 2022

INFORMATION SCIENCE AND ENGINEERING

III & IV SEMESTER B.E

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
Bangalore – 560054.

About the Institute:

Dr. M. S. Ramaiah a philanthropist, founded 'Gokula Education Foundation' in 1962 with an objective of serving the society. M S Ramaiah Institute of Technology (MSRIT) was established under the aegis of this foundation in the same year, creating a landmark in technical education in India. MSRIT offers 17 UG programs and 15 PG programs. All these programs are approved by AICTE. All eligible UG and PG programs are accredited by National Board of Accreditation (NBA). The institute is accredited with **'A^{+}' grade by NAAC in March 2021** for 5 years. University Grants Commission (UGC) & Visvesvaraya Technological University (VTU) have conferred Autonomous Status to MSRIT for both UG and PG Programs since 2007. The institute is a participant to the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. The institute has 380 competent faculty out of which 60% are doctorates. Some of the distinguished features of MSRIT are: State of the art laboratories, individual computing facility for all faculty members, all research departments active with sponsored funded projects and more than 300 scholars pursuing Ph.D. To promote research culture, the institute has established Centre of Excellence for Imaging Technologies, Centre for Advanced Materials Technology, Centre for Antennas and Radio Frequency systems (CARFS), Center for Cyber Physical Systems & Schneider Centre of Excellence. **M S Ramaiah Institute of Technology has obtained "Scimago Institutions Rankings" All India Rank 65 & world ranking 578 for the year 2020.**

The Entrepreneurship Development Cell (EDC) and Section 8 company "Ramaiah Evolute" have been set up on campus to incubate startups. **M S Ramaiah Institute of Technology secured All India Rank 8th for the year 2020 for Atal Ranking of Institutions on Innovation Achievements (ARIIA), by MoE, Govt. of India.** MSRIT has a strong Placement and Training department with a committed team, a good Mentoring/Proctorial system, a fully equipped Sports department, large air-conditioned library with good collection of book volumes and subscription to International and National Journals. The Digital Library subscribes to online e-journals from Elsevier Science Direct, IEEE, Taylor & Francis, Springer Link, etc. MSRIT is a member of DELNET, CMTI and VTU E-Library Consortium. MSRIT has a modern auditorium and several hi-tech conference halls with video conferencing facilities. The institute has excellent hostel facilities for boys and girls. MSRIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association.

As per the National Institutional Ranking Framework (NIRF), MoE, Government of India, M S Ramaiah Institute of Technology has achieved 65th rank among 1143 top Engineering institutions of India for the year 2021 and is 1st amongst the Engineering colleges affiliated to VTU, Karnataka.

About the Department:

Information Science and Engineering department is 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 also started M.Tech program in Software Engineering in the year 2004 and has been recognized as R&D center by VTU in 2012. The department is accredited by the NBA in 2001, 2004, 2010, 2015 and reaccredited in 2018 under Tier-1 till 2022. Department has highly qualified and motivated faculty members and well equipped state of the art laboratories. All faculty members are involved in research and technical papers publications in reputed journals, conferences across the world. Strong collaboration with industries and high profile institutions is in place for curriculum updates, more hands on training, practical's, project based learning, EPICS, expert lectures, partial course deliveries by industry experts and student interns to enhance the skills in emerging areas to keep an inclusive and diverse academic environment. Department is regularly conducting seminars, conferences and workshops for students and academicians in the emerging areas of Information Technology. Introduced EPICS in senior projects. Some of the laboratories have also been set up in collaboration with industries such as Intel, Microsoft, Apple, SECO, Honeywell, EMC^2, NVIDIA, IBM, Green Sense Werks, Tech Machinery Labs, Sesovera Tech Pvt. Ltd., and Ramaiah Medical College (Emergency department). Also, an echo system is built to initiate start-ups at the department level along with the mentorship. All the above potential activities have led to high profile placements, motivation to become an entrepreneur, and encouragement for higher learning.

VISION OF THE INSTITUTE

To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio-economic needs

MISSION OF THE INSTITUTE

MSRIT shall meet the global socio-economic needs through

- Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
- Establishing research clusters in emerging areas in collaboration with globally reputed organizations
- Establishing innovative skills development, techno-entrepreneurial activities and consultancy for socio-economic needs

QUALITY POLICY

We at MS Ramaiah Institute of Technology 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 THE 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 THE 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.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Become competent Information Technology professionals with continuous progress in career or learning.

PEO2: Enhance the skills in developing computing systems using modern tools and technologies.

PEO3: Function effectively as professionals in a team environment or individually.

PROGRAM OUTCOMES (POs)

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

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

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

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

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

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

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

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

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

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

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

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

PROGRAM SPECIFIC OUTCOMES (PSOs)

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 Higher Learning, ability to exchange knowledge and Continue learning advances in the field of Information Technology.

Semester wise Credit Breakdown for B E Degree Curriculum

Batch 2020-24

Semester / Course Category	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth	Total Credits
Basic Sciences (BSC)	9	8	4	4					**25**
Engineering Sciences (ESC)	11	10							**21**
Humanities, Social Sciences and Management (HSMC)		2			3		3		**8**
Professional Courses – Core (PCC)			21	21	15	11	10		**78**
Professional Courses– Elective (PEC)					3	6	6	3	**15**
Other Open Elective Courses (OEC)					3	3			**6**
Project Work (PROJ), Internship (IN)						4	1	14	**22**
Total Credits	**20**	**20**	**25**	**25**	**24**	**24**	**20**	**17**	**175**

SCHEME OF TEACHING
III SEMESTER

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	IS31	Engineering Mathematics - III	BS	3	1	0	4	5
2.	IS32	Computer Organization and Architecture	PC-C	4	0	0	4	4
3.	IS33	Discrete Mathematical Structures	PC-C	3	1	0	4	4
4.	IS34	Data Communications	PC-C	3	0	0	3	3
5.	IS35	Data Structures	PC-C	4	0	0	4	4
6.	IS36	Object Oriented Programming with Java	PC-C	4	0	0	4	4
7.	ISL37	Data Structures Lab	PC-C	0	0	1	1	1
8.	ISL38	Object Oriented Programming with Java Lab	PC-C	0	0	1	1	1
			Total	**21**	**2**	**2**	**25**	**26**

L – Lecture (one hour) T - Tutorial (Two hours) P - Practical (Two hours)

Note:

1. The Non Credit Mandatory Course, Additional Mathematics – I is prescribed for III Semester Lateral Entry Diploma students admitted to III Semester of BE Program. The student shall register for this course along with other III semester courses. The students shall attend classes for the course during the semester and complete all formalities of attendance and CIE to appear for SEE. This Course shall not be considered for vertical progression, but completion of the course shall be mandatory for the award of the degree.

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1	AM31	Additional Mathematics - I	BSC	0	0	0	0	3

2. **AICTE Activity Points to be earned by students admitted to BE program (For more details refer to Chapter 6, AICTE, Activity Point Program, Model Internship Guidelines):**
   Every regular student, who is admitted to the 4 year degree program, is required to earn 100 activity points in addition to the total credits earned for the program. Students entering 4 years Degree Program through lateral entry are required to earn 75 activity points in addition to the total credits earned for the program. The activity points earned by the student shall be reflected on the students 8th Semester grade card. The activities to earn the points can be spread over the duration of the course. However, minimum prescribed duration should be fulfilled. Activity Points (non-credit) have no effect on SGPA/CGPA and shall not be considered for vertical progression. Incase student fail to earn the prescribed activity points, Eight semester Grade Card shall be issued only after earning the required activity Points. Students shall be eligible for the award of degree only after the release of the Eight Semester grade card.

SCHEME OF TEACHING
IV SEMESTER

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	IS41	Engineering Mathematics - IV	BS	3	1	0	4	5
2.	IS42	Operating Systems	PC-C	3	0	0	3	3
3.	IS43	Operations Research	PC-C	3	1	0	4	4
4.	IS44	Finite Automata and Formal Languages	PC-C	3	1	0	4	4
5.	IS45	Design and Analysis of Algorithms	PC-C	4	0	0	4	4
6.	IS46	Microcontrollers	PC-C	4	0	0	4	4
7.	ISL47	Design and Analysis of Algorithms Lab	PC-C	0	0	1	1	1
8.	ISL48	Microcontrollers Lab	PC-C	0	0	1	1	1
Total				**20**	**3**	**2**	**25**	**26**

* Non Credit Mandatory Course L – Lecture (one hour) T - Tutorial (Two hours) P - Practical (Two hours)

Note:

1. The Non Credit Mandatory Course, Additional Mathematics – II is prescribed for IV Semester Lateral Entry Diploma students admitted to BE Program. The student shall register for this course along with other IV semester courses. The students shall attend classes for the course during the semester and complete all formalities of attendance and CIE to appear for SEE. This Course shall not be considered for vertical progression, but completion of the course shall be mandatory for the award of the degree.

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1	AM41	Additional Mathematics - II	BSC	0	0	0	0	3

2. **AICTE Activity Points to be earned by students admitted to BE program (For more details refer to Chapter 6, AICTE, Activity Point Program, Model Internship Guidelines):**
   Every regular student, who is admitted to the 4 year degree program, is required to earn 100 activity points in addition to the total credits earned for the program. Students entering 4 years Degree Program through lateral entry are required to earn 75 activity points in addition to the total credits earned for the program. The activity points earned by the student shall be reflected on the students 8th Semester grade card. The activities to earn the points can be spread over the duration of the course. However, minimum prescribed duration should be fulfilled. Activity Points (non-credit) have no effect on SGPA/CGPA and shall not be considered for vertical progression. Incase student fail to earn the prescribed activity points, Eight semester Grade Card shall be issued only after earning the required activity Points. Students shall be eligible for the award of degree only after the release of the Eight Semester grade card.

III Semester

ENGINEERING MATHEMATICS - III

Course Code: IS31 **Credit: 3:1:0**
Prerequisite: Calculus **Contact Hours:42L+14T**
Course Coordinator: Dr. N.L Ramesh & Dr. Uma M

Course Content

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 linear, quadratic and geometric curves, Correlation and Regression.

Unit II

Linear Algebra I: Elementary transformations on a matrix, Echelon form of a matrix, Rank of a matrix, Consistency of system of linear equations, Gauss elimination and Gauss – Seidel 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 square matrices, Solution of system of ODEs using matrix method.

Unit-III

Linear Algebra II: Symmetric matrices, Orthogonal diagonalization and Quadratic forms, Vector Spaces, Linear combination and span, Linearly independent and dependent vectors, Basis and Dimension, Linear transformations, Composition of matrix transformations, Rotation about the origin, Dilation, Contraction and Reflection, Kernel and Range, Change of basis.

Unit-IV

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

Unit-V

Fourier Transforms: Infinite Fourier transform, Fourier sine and cosine transform, Properties, Inverse transform. Limitations of Fourier transforms and need of Wavelet transforms.

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.

Text Books:

1. Erwin Kreyszig-Advanced Engineering Mathematics-Wiley-India publishers- 10th edition-2015.
2. B.S. Grewal - Higher Engineering Mathematics - Khanna Publishers – 44th edition-2017.

References:

1. David C. Lay, Steven R. Lay and Judi J. Mc. Donald – Linear Algebra and its Applications – Pearson – 5th edition – 2015.
2. Peter V. O'Neil – Advanced Engineering Mathematics – Cengage learning – 7th edition – 2011.
3. Gareth Williams – Linear Algebra with Applications, Jones and Bartlett Press – 9th edition – 2017.
4. Christian Blatter – Wavelets, CRC Press –Published – 2018.

Course Outcomes (COs):

At the end of the course, students will be able to

1. Apply numerical techniques to solve engineering problems and fit a least squares curve to the given data. (PO-1,2) (PSO-2)
2. Test the system of linear equations for consistency and solve system of ODE's using matrix method. (PO-1,2) (PSO-2)
3. Diagonalize a given matrix orthogonally and find kernel and range of linear transformation. (PO-1,2) (PSO-2)
4. Construct the Fourier series expansion of a function/ tabulated data. (PO-1,2) (PSO-2)
5. Evaluate Fourier transforms of given functions and use Z-transforms to solve difference equations. (PO-1,2) (PSO-2)

COMPUTER ORGANIZATION AND ARCHITECTURE

Course Code: IS32 **Credit: 4:0:0**
Prerequisite: Basic Electronics **Contact Hours: 56L**
Course Coordinator: Mrs. Prathima M N

Course Content

UNIT I

Introduction: Basics of Computer Organization and Architecture, the general functions and structure of a digital computer, Performance Assessment-clock speed and Instructions per second, Benchmarks, Amdahl's Law and Little's Law.
The Computer System: Computer Components, Computer Functions, Interconnection structures and Bus Interconnection.

UNIT II

Computer Memory System overview: Cache Memory Principles, Terminology and Definitions of Internal memory and External Memory Types
Input and Output Device: I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, External Interconnection Standards

UNIT III

The Arithmetic and Logic Unit: Integer Representation: Sign Magnitude Representations, 2's complement, Integer arithmetic operations: Booth's Algorithm for Twos Complement Multiplication, Floating-Point Representation, Floating point Arithmetic, Combinational and Sequential Circuits.

UNIT IV

The Central Processing Unit: Machine Instruction Characteristics, Terminology and definitions of types of Operands and Operations, Addressing Modes, Processor and Register organization, instruction cycle.
Case Study: Case studies of some contemporary advanced architecture for processors of families like Intel, AMD, IBM etc.

UNIT V

Multiple Processor Organizations: A Taxonomy of Parallel Processor Architectures, Symmetric Multiprocessors, Cache Coherence and the MESI protocol, Multithreading and Chip Multiprocessors
The Control Unit: The Control Unit Operations - Micro-operations, Control of the Processor, Hardwired Implementation and basic concepts of Micro programmed Control

Text Book:

1. William Stallings, "Computer Organization and Architecture, Designing for Performance", 10e Pearson, 2016.

References:

1. Patterson D. A., Hennessy J. L., "Computer Organization and Design", Morgan Kaufmann, 5e 2014.
2. M. Murdocca & V. Heuring, "Computer Architecture and Organization: An integrated Approach" Wiley, 2007.

Course Outcomes (COs):

At the end of the course, students will be able to -

1. Describe the general functions and structure of a digital computer and the concept of interconnection within a computer system. (PO-1,2,7,12) (PSO-1,2,3)
2. Discuss the main characteristics of computer memory systems and I/O systems. (PO-1, 2, 12) (PSO-1, 2)
3. Discuss Arithmetic Logic Unit implementation concepts including combinational circuits, sequential circuits and programmable logic devices. (PO-1,2,3) (PSO-1,2)
4. Describe the internal structure and functioning of Central Processing Unit including instruction execution. (PO-1,2,4,10) (PSO-3)
5. Describe modern computer architectures and discuss how micro-operations are organized to control processor functions. (PO-1,2,5,12) (PSO-1,2,3)

DISCRETE MATHEMATICAL STRUCTURES

Course Code: IS33 **Credit: 3:1:0**
Prerequisite: Nil **Contact Hours:42L+14T**
Course Coordinator: Mrs. Prathima M N

Course Content

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: Types of Functions, Functions for computer science, Permutation functions, Order relations and structures: Partially ordered sets, Extremal elements of partially ordered sets, Lattices.

UNIT-IV

Graphs: Graphs and graph models, Graph terminology and special types of graphs, Representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths.

UNIT-V

Semigroups and Groups: Binary operations revisited: Tables, Semigroups: Sub semigroup, Submonoid, Isomorphism, Homomorphism, Group.
Mathematical structures: Rings, Fields and Fermat's little theorem.

Tutorial Sessions:

1. Problems solving based on sets, operation on sets.
2. Representation of different formulae in Sequence and identifying regular expressions.
3. Performing different operations using Logical operators and verifying different statements using Mathematical induction.
4. Problems on counting techniques.
5. Working out problems in Relations and digraph.
6. Problems related to operations on relations, transitive closure and Warshall's algorithm.
7. Identify types of Functions, Functions for Computer Science.
8. Usage of Partially ordered relations and structures in terms of Hasse diagram and Topological Sorting.
9. Identify Hasse diagram represent lattices and Usage of types of lattices.
10. Terminologies in graphs and their models.

11. Problems related to Euler, Hamilton paths/circuits, semigroups.
12. Verifying different properties representing Group.
13. Problems on other mathematical structures.
14. Identifying noise in coding of binary information and detection of errors.

Text Books:

1. Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, Discrete Mathematical Structures, 6th 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 (COs):

At the end of the course, students will be able to-

1. Apply the properties of set theory and logical reasoning to verify the correctness of mathematical statements. (PO-1,2) (PSO-1)
2. Employ the notation of relation and principle of counting in problem solving. (PO-1,2) (PSO-1)
3. Illustrate the concepts of functions, partially ordered sets, lattices for computer science. (PO-1,2) (PSO-1)
4. Apply the concepts of graphs theory in computing problems. (PO-1,2,3) (PSO-1)
5. Apply the concepts of groups for understanding mathematical structures. (PO-1,2) (PSO-1)

DATA COMMUNICATIONS

Course Code: IS34 **Credit: 3:0:0**
Prerequisite: Basic Electronics **Contact Hours: 42L**
Course Coordinator: Mr. Suresh Kumar K R

Course Content

UNIT-I

Data communication Fundamentals: Introduction, components, Data Representation, Data Flow; Networks – Network criteria, Physical Structures, Network Models, Categories of networks (with problems to solve); Protocols, Standards, Standards organization; The Internet – Brief history, Internet today; **Network Models -** Layered tasks; The OSI model – Layered architecture, Peer-to-Peer Process, Encapsulation; Layers in the OSI model; TCP/IP Protocol suite; Addressing.

UNIT-II

Digital Transmission Fundamentals (with problems to solve): Analog & Digital data, Analog & Digital signals (basics); Transmission Impairment – Attenuation, Distortion and Noise; Data rate limits – Nyquist Bit Rate, Shannon Capacity; Performance, **Digital Transmission (with problems to solve):** Digital-to-Digital conversion - Line coding, Line coding schemes (unipolar, polar, bipolar); Analog-to-Digital conversion - PCM.

UNIT-III

Error detection & correction (with problems to solve): 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); Noisy channels (Stop-and-wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ, Piggybacking).

UNIT-IV

HDLC – Transfer modes, frames: **Point-to-Point Protocol** – Framing, transition phases; **Multiple Access (with problems to solve):** Random Access (Aloha, CSMA, CSMA/CD, CSMA/CA), Controlled Access (Reservation, Polling, Token Passing), Channelization (FDMA, TDMA, CDMA)

UNIT-V

Wired LANs: IEEE standards; Standard Ethernet; **Wireless LANs:** IEEE 802.11 - Architecture, MAC sublayer, addressing mechanism, Bluetooth and its architecture; Connecting devices, Backbone networks, Virtual LANs

Text Books:

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

Reference:

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 (COs):

At the end of the course, students will be able to

1. Distinguish different communication models / protocol stacks (OSI & TCP/IP) and analyze the usage of appropriate network topology for a given scenario. (PO-1, 2, 3) (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) (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-1, 2, 3) (PSO-2)
4. Use the different strategies of multiple access to achieve better network efficiency and analyze the network performance. (PO-1, 2) (PSO-1, 2)
5. Illustrate the IEEE standards for wired, wireless LANs and their connecting devices. (PO-3, 10) (PSO-2)

DATA STRUCTURES

Course Code: IS35 **Credits: 4:0:0**
Prerequisites: Fundamentals of Computing **Contact Hours: 56L**
Course Coordinator: Dr. Mydhili K Nair

Course Content

UNIT-I

Introduction to Data Structures: Definition – Basic Concepts on Arrays, Stacks, Queues, Trees and Linked Lists. **Introduction to File Structures:** Comparing Data Structures with File Structures, Short history of file structure design, Differentiating between Physical files and logical files **Recursion**: Recursive definition and processes using Stacks and Trees, Examples – Factorial,Tower of Hanoi;

UNIT-II

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-III

The Stack: Definition, Representation, Basic operations of stack (PUSH and POP), its implementation **Applications of stack:** Conversion from Infix to Postfix, Evaluation of Postfix expression; **The Queues:** Definition, Representation, Primitive operations of queue and its implementation; **Types of Queues:** Circular queues and Priority queues

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

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, YedidyahLangsam 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

References:

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 (COs):

At the end of the course, student will be able to -

1. Explain the working principle of data structures, their applications and figure out which data structure is best suited for a given problem statement.(PO-1, 2, 3, 6, 7) (PSO-1,2)
2. Describe the concept of static & dynamic memory allocation, Linked List as a data structure for dynamic memory allocation and learn their design and implementation. .(PO-1, 2, 3, 6, 7) (PSO-1,2)
3. Develop and implement the basic data structures namely Stacks, Queues, Trees, Linked Lists and their various variants. .(PO-1, 2, 3, 6, 7) (PSO-1,2)
4. Develop Stack and Queues using the linear data structure linked lists, Design and develop non-linear data structures such as Binary Trees and Binary Search Trees .(PO-1, 2, 3, 6, 7) (PSO-1,2)
5. Appreciate the need for file structures such as B-Trees and B+ Trees, learn the basics of operating them and describe concepts such as hashing and its various techniques. .(PO-1, 2, 3, 6, 7) (PSO-1,2)

OBJECT ORIENTED PROGRAMMING WITH JAVA

Course Code: IS36 **Credit: 4:0:0**
Prerequisite: Fundamentals of Computing **Contact Hours: 56L**
Course Coordinator: Dr. S R Mani Sekhar

Course Content:

UNIT-I

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. **Introducing Classes:** Class Fundamentals, Declaring Objects, A Closer Look at new, Assigning Object Reference Variables, Introducing Methods, Constructors, Parameterized Constructors, The this Keyword, Instance Variable Hiding, Garbage Collection, The finalize Method, A Stack Class. **UML Notations** for Classes and Objects.

UNIT-II

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, Arrays Revisited, Introducing Nested and Inner Classes, Exploring the String Class, Using Command-Line Arguments, Varargs: Variable-Length Arguments. **Inheritance:** Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding. **UML Notations** for Inheritance, Composition and Aggregation

UNIT-III

Inheritance: Dynamic Method Dispatch, Why Overridden Methods? Using Abstract Classes, Using final with Inheritance, The Object Class. Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Defining an Interfaces, Default Interface Methods, Use static Methods in an Interface, Final Thoughts on Packages and Interfaces. UML Notations for Packages and Interfaces

UNIT-IV

Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, multiple catch Clauses, Nested try Statements, throw, throws, finally. Exception Handling: Java's Built-in exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Three Recently Added Exception Features, Using Exceptions.

UNIT-V

Type Wrappers: Character, Boolean, Numeric type wrappers. Autoboxing: Autoboxing and Methods, Autoboxing / Unboxing occur in expressions, Autoboxing/Unboxing Boolean and Character values, Autoboxing / Unboxing helps prevents errors. **The Collections Framework:** Collections Overview, The Collection Interfaces: the collection interface, the List interface, the Set interface. The Collection Classes: The ArrayList Class, The LinkedList Class, The HashSet Class. Accessing a Collection via an Iterator -Using an Iterator, The For-Each Alternative to Iterators.

Text Books:

1. Herbert Schildt, "Java: The Complete Reference", 9th Edition, McGraw Hill
2. Michael Blaha, James Rambaugh, "Object-Oriented Modeling and Analysis with UML", Pearson, 2nd Edition, 1st Impression

Reference:

1. E. Balagurusamy; Programming with Java, McGraw-Hill; Sixth edition.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Write programs using Java-specific programming constructs and Object Oriented way of programming through Classes and Objects. (PO- 1, 2, 3, 5) (PSO- 1, 2)
2. Design solutions to real-world problems using UML Notations for Classes, Objects, Inheritance, Composition and Aggregation. (PO- 1, 2, 3, 5) (PSO- 1, 2)
3. Write programs using concepts using dynamic memory dispatch concept of inheritance, packages and interfaces. (PO- 1, 2, 3, 5) (PSO- 1, 2)
4. Apply the concepts of Exception Handling nuances to solve a given real-world problem. (PO- 1, 2, 3, 5) (PSO- 1, 2)
5. Use the Java's Collection framework to solve computing real-world problems. (PO- 1, 2, 3, 5) (PSO- 1, 2)

DATA STRUCTURES LABORATORY

Course Code: ISL37 **Credit : 0:0:1**

Prerequisite: Fundamentals of Computing Laboratory **Contact Hours:14P**

Course Coordinator: Mrs. Shruthi G.

Laboratory Experiments:

Part - A

Implement the following experiments:

1. Primitive operations on Stacks using arrays.
2. Primitive operations on Linear Queue using arrays.
3. Evaluation of a valid Postfix expression using stacks.
4. Implement the following using recursion
    a) Tower of Hanoi
    b) Binary search
    c) GCD
5. Circular queue using arrays.
6. Stack using Singly Linked List.
7. Queues using Singly Linked List.
8. To reverse a given Singly Linked List.
9. Implement the following using Circular Linked List.
    a) Stack
    b) Queues
10. Creation and display of Binary Search Tree.

Part - B

Implement the following experiments:

1. Conversion of a valid Infix expression to Postfix Expression using stacks. Program should support for both parenthesized and parenthesize free expressions with the operators: +, -, *, /, % (Remainder), ^(Power) and alphanumeric operands.
2. Priority queue using arrays.
3. Singly Linked List with the following operations:
    a. Inserting a node (Any desired position)
    b. Deleting a node (Any desired position)
    c. Display
4. Circular Linked List with the following operations:
    a. Inserting a node (Any desired position)
    b. Deleting a node (Any desired position)
    c. Display
5. Doubly Linked List with the following operations:
    a. Inserting a node (Any desired position)
    b. Deleting a node (Any desired position)
    c. Display
6. Represent and evaluate a given Polynomial using Singly Linked List.
7. To insert a given element into an ordered singly linked list or Doubly Linked List.

8. To delete every second node from Singly Linked list.
9. Binary Tree operations:
    a. Creation
    b. Traversal (Inorder, Preorder and Postorder)
10. Creation of Expression tree and evaluate it

Reference:

1. Aaron M. Tanenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C".
2. Horowitz and Sahani. "Fundamentals of Data Structures", Galgotia Publication Pvt Ltd., New Delhi.
3. Behrouz A. Forouzan and Richard F. Gilberg, "Computer Science A Structured Programming Approach using C", Second Edition, Thomson Publications.
4. R. Kruse, "Data Structures and Program Design in C", Pearson Education.

Course Outcomes (COs):

At the end of the course, students will be able to

1. Design and implement the concepts of data structures. (PO –1, 2, 3, 4, 5, 12) (PSO – 1, 2)
2. Demonstrate the working principle of different types of data structures and their applications. (PO – 1, 3, 10) (PSO – 1, 3)
3. Analyze the results and produce substantial written documentation. (PO – 1, 4, 10, 12) (PSO–1, 2, 3)

OBJECT ORIENTED PROGRAMMING WITH JAVA LABORATORY

Course Code: ISL38 **Credit : 0:0:1**

Prerequisite: Fundamentals of Computing **Contact Hours:14P**

Course Coordinator: Koushik S

Programs

1. Write Java programs
    a. To print fibonacci series without using recursion and using recursion.(concept of loops, data types)
    b. To check prime numbers.
    c. To sort an array elements using bubble sort algorithm.
2. Create a class called account with the data members (Accno – integer, name String, Phone_No: integer, balance_amt:float), and following methods :
    a. getinput() to get input from the user
    b. Deposit() method which takes the amount to be deposited in to his/her account and do the calculation.
    c. Withdraw() method which gets the amount to be withdrawn from his/her account.
    d. Print the appropriate results.
3. Define a Stack class to implement the stack data structure. Include constructors to perform initialization, method push to push an element into the stack, method pop to remove an element from the stack and display method to display the elements of the stack.
4. Define a class Complex with data members as two real numbers, constructors for initialization these numbers, methods to add, subtract and multiply 2 complex numbers.
5. Write a java program to read 2 matrices and place the product in a third matrix . Use constructors and suitable methods.
6. Write a java program to work with strings.
    a. Extract a portion of the string and print it. Variable m indicates the amount of characters to be extracted from the string starting from the nth position.
    b. Read a text and count all the occurrences of a particular word.
    c. Replace a substring in the given string.
    d. Rearrange the string and rewrite in alphabetical order.
    e. Compare two strings ignoring case.
    f. Concatenate two strings.
7. Create a Personal class to hold the personal details of an person such as name, age, education, salary- (basic, da, hra), years of experience, number of loans and loan amount. Write constructors to assign values to the data members. Include an
    a. isEligible() method to indicate whether the person is eligible for loan,
    b. taxPay() method to indicate the amount of tax to be paid,
    c. isEligiblePromotion() to indicate whether the person is eligible for a promotion.
    d. Display () method to display the details.

    Enter the details of n employees and indicate their eligibility and the tax to be paid.

8. Create a Circle class with following members.
    a. A data member that stores the radius of a circle.
    b. A constructor function with an argument that initializes the radius
    c. A function that computes and returns area of a circle

Create two derived classes Sector and Segment that inherit the Circle class. Both classes inherit radius and the function that returns the circle's area from Circle.

In addition to the members inherited from Circle, Sector and Segment have some specific members as follows:

Sector

1. A data member that stores the control angle of a sector (in radians)
2. A constructor function with arguments that initialize radius and angle
3. A function that computes and returns the area of a sector

Segment

1. A data member that stores the length of a segment in a circle
2. A constructor function with arguments that initialize radius and length
3. A function that computes and returns the area of a segment

Create the main () function to instantiate an object of each class and then call appropriate member functions to compute and return the area of a circle, sector and segment.

Note:

a. Area_of_circle = pi * r^2
b. Area_of_Sector=(1/2) r^2 *θ
c. Area_of_segment= r^2 *((r-h)/r) – (r-h) (2rh-h^2)^1/2

Where r is the radius of a circle, θ is the central angle of a sector in radians, h is the length of a segment and ((r-h)/r) is in radians.

9. Write a Java Program that does the following related to Inheritance:
    a. Create an abstract class called Vehicle which contains the „year_of_manufacture" data member and two abstract methods „getData()" and „putData()" with a constructor.
    b. Create two derived classes "TwoWheeler" and "FourWheeler" and implement the abstract methods. Make "FourWheeler" as final class.
    c. Create class „MyTwoWheeler" which is a sub-class of "TwoWheeler" and demonstrate the use of super keyword to initialize data members of "MyTwoWheeler".
10. Define an interface "Department" with methods to readdata() and printdata(), print_number_designations(), number_research_consultancy_projs().

Define a "Faculty" class with members name, designation, age, years of experience, joining_date and subjects_handled.

    a. In package ISE define the "ISE_department" class that implements the "Department" interface, accepts n faculty details and define all the methods. Raise a user defined exception "AgeException" if the age of the faculty is > 58.
    b. In the default package define a "MainClass" which uses the methods of the above classes and also displays those faculty details whose years of experience is greater than or equal to 20.

11. Write a Java Program that does the following related to Packages and Interfaces , Exception Handling:
    a. Create an interface Student which gets the name and branch of a student.
    b. Create a package called "StudentPackage" which has a user-defined class RegisterStudent.
    c. If a student registers above 30 credits for the semester, the method should throw a user-defined exception called "CreditLimit" and display an appropriate message.
    d. Create another package called "ResultPackage" which displays the grade for the subject registered for particular semester and the SGPA. If SGPA is above 10 then throws an InvalidSGPA user defined exception.
    e. In the StudentPackage , collect the marks of all the subjects in 4 semesters and calculate SGPA and CGPA.
12. a. Write a java program to implement queues of Strings using an ArrayList class of the Collection framework.
    b. Create a linked list of names (String type). Use an Iterator to traverse through the list and print those names whose length is < 5.

Text Books

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

Course Outcomes (COs):

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

1. Design and develop programs using Java-specific programming constructs. (PO – 1, 2, 3, 5) (PSO- 1, 2)
2. Design solutions to variety of problems using Classes and Objects. (PO- 2, 3, 4, 5) (PSO- 1, 2)
3. Design, reuse and develop solutions using the concepts of Inheritance, Packages and Interfaces. (PO 2, 3, 4, 5) (PSO 1, 2)
4. Identify exceptions in a program and handle them. (PO -2, 3, 4, 5) (PSO- 1, 2)
5. Use the collection framework to solve real world problems. (PO- 2, 3, 4, 5) (PSO- 1, 2)

ADDITIONAL MATHEMATICS - I

Course Code: AM31 **Credit: 0:0:0**

Prerequisite: Nil **Contact Hours:40**

Course Coordinator: Dr. N. L. Ramesh

Course content

UNIT-I

Differential Calculus-I: Successive differentiation, n^{th} derivatives of some standard functions, Leibnitz theorem, Polar curves. Angle between the radius vector and the tangent, angle between curves, length of the perpendicular from pole to the tangent, pedal equations. Taylor's and Maclaurin's expansions.

UNIT-II

Integral Calculus: Introduction, Reduction formula, Reduction formula for $\int sin^n x\, dx$, Reduction formula for $\int cos^n x\, dx$, Reduction formula for $\int sin^n x cos^m x\ dx$, Evaluation of double and triple integrals.

UNIT-III

Vector Algebra: Scalar and vectors. Vector addition and subtraction. Multiplication of vectors (Dot and Cross products). Scalar and vector triple product-simple problems. Vector functions of a single variable. Derivative of a vector function, geometrical interpretation. Velocity and acceleration.

UNIT-IV

Vector Differentiation: Scalar and vector fields, gradient of a scalar field, directional derivative, divergence of a vector field, solenoidal vector, curl of a vector, irrational vector, Laplace's operator. Vector identities connected with gradient, divergence and curl.

UNIT-V

First Order Differential Equations: Solution of first order and first degree differential equations, variable separable methods, homogeneous equations, linear and Bernoulli's equations, exact differential equations.

Text Book:

1. B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers, 44th edition, 2017.
2. Erwin Kreyszig –Advanced Engineering Mathematics, Wiley publication, 10th edition, 2015.

References:

1. H.K. Dass – Higher Engineering Mathematics – S Chand Publications - 1998.
2. B.V. Ramana – Engineering Mathematics – Tata McGrawHill Publishing Co. Ltd. – New Delhi – 2008.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Find the length of the perpendicular from pole to tangent and determine the series expansion of differentiable functions (PO-1, 2)
2. Evaluate multiple integrals (PO-1, 2)
3. Analyze and solve problems related to Vector Algebra. (PO-1, 2)
4. Apply vector differentiation to identify solenoidal and irrotational vectors. (PO-1, 2)
5. Solve the first order and first degree ordinary differential equations. (PO-1, 2)

IV Semester

ENGINEERING MATHEMATICS-IV

Course Code: IS41 **Credit: 3:1:0**
Prerequisite: Calculus & Probability **Contact Hours: 42L+14T**
Course Coordinator: Dr. N.L Ramesh & Dr. Uma M

Course Content:

Unit-I

Finite Differences and Interpolation: Forward, Backward differences, Interpolation, Newton-Gregory Forward and Backward Interpolation formulae, Lagrange's interpolation formula and Newton's 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's 1/3rd rule, Simpson's 3/8th rule.

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, Uniform distribution, Exponential distribution, Gamma distribution and Normal distribution.

Unit III

Joint probability distribution: Joint probability distribution (both discrete and continuous), Conditional probability, 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, Symbolic representation of a queuing model, Single server Poisson queuing model with infinite capacity (M/M/1 : ∞ /FIFO), when $\lambda_n = \lambda$ and $\mu_n = \mu$ $(\lambda < \mu)$, Performance measures of the model, Single server Poisson queuing model with finite capacity (M/M/S : N/FIFO), Performance measures of the model, Derivations of difference equations and expressions for L_s, L_q, W_s, W_q of M/M/1 queuing model with finite and infinite capacity, Multiple server Poisson queuing model with infinite capacity (M/M/S : ∞ /FIFO), when $\lambda_n = \lambda$ for all n, $(\lambda < S\mu)$, Multiple server Poisson queuing model with finite capacity (M/M/S : N/FIFO), Introduction to M/G/1 queuing model.

Unit-V

Sampling and Statistical Inference: Sampling of Gaussian distributions, Concepts of standard error and confidence interval, Central Limit Theorem, Type-1 and Type-2 errors, Level of significance, One tailed and two tailed tests, Z-test: for single mean, for single proportion, for difference between means, Student's t –test: for single mean, for difference between two means, F – test: for equality of two variances, Chi-square test: for goodness of fit, for independence of attributes.

Text Books:

1. R.E. Walpole, R. H. Myers, R. S. L. Myers and K. Ye – Probability and Statistics for Engineers and Scientists – Pearson Education – Delhi – 9th edition – 2012.
2. B.S. Grewal - Higher Engineering Mathematics - Khanna Publishers – 44th edition-2017.
3. T. Veerarajan- Probability, Statistics and Random processes – Tata McGraw-Hill Education – 3rd edition -2017.

References:

1. Erwin Kreyszig - Advanced Engineering Mathematics-Wiley-India publishers- 10th edition-2015.
2. Sheldon M. Ross – Probability models for Computer Science – Academic Press, Elsevier– 2009.
3. Murray R Spiegel, John Schiller & R. Alu Srinivasan – Probability and Statistics – Schaum's outlines -4th edition-2012.
4. Kishore S. Trivedi – Probability & Statistics with Reliability, Queuing and Computer Science Applications – John Wiley & Sons – 2nd edition – 2008.

Course Outcomes (COs):

At the end of the course, students will be able to

1. Find functional values, derivatives, areas and volumes numerically from a given data. (PO-1,2) (PSO-2)
2. Analyze the given random data and their probability distributions. (PO-1,2) (PSO-2)
3. Calculate the marginal and conditional distributions of bivariate random variables and determine the parameters of stationary random processes. (PO-1,2) (PSO-2)
4. Use Markov chain in prediction of future events and in queuing models. (PO-1,2) (PSO-2)
5. Choose an appropriate test of significance and make inference about the population from a sample. (PO-1,2) (PSO-2)

OPERATING SYSTEMS

Course Code: IS42 **Credits: 3:0:0**
Prerequisite: Nil **Contact Hours: :42L**
Course Coordinator: Mrs. Shruthi G

Course Content

UNIT-I

Introduction: Operating System Structure, Operating System Operations, Process Management, Memory Management, Storage Management, Protection and Security; Process Concept: Process Scheduling, Operations on Processes, Interprocess Communication; CPU Scheduling: Scheduling Criteria, Scheduling algorithms Multiple processor scheduler.

UNIT-II

Synchronization: The Critical Section Problem, Hardware and software solution for critical section problem, Synchronization Scheduling Algorithms, Mutex locks, Semaphores, Classical Problems of Synchronization, Monitors; Deadlocks: System Model, Deadlock Characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance and detection, Recovery from Deadlock.

UNIT-III

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation; Virtual Memory Management: Background, Demand Paging, Copy on Write, Page Replacement, Allocation of frames, Allocating Kernel Memory.

UNIT-IV

File System: File Concept, Access Methods, Directory Structure, Protection; Implementing File Systems: File System Structure, File System Implementation, Directory Implementation, Allocation Methods, Free Space Management, Efficiency and Performance, Recovery; Secondary-Storage Structure: Disk structure, Disk Attachment, Disk Scheduling Methods, Disk management, Swap-Space Management.

UNIT-V

System Protection: Goals of Protection, Principle of Protection, Access Matrix, Domain of Protection, Access Matrix and its implementation, Case Study on The Linux System: Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management File Systems, Inter process Communication. Case Studies on Linux and Macintosh.

Text Book:

1. Abraham Silberschatz, Peter B Galvin, Gerg Gagne, Operating Systems Concepts, 9th Edition, 2016 India Edition, Wiley Publications.

References:

1. Dhananjay M Dhamdhere, Operating Systems: A Concept-Based Approach, 3rd Edition, McGraw Hill Education, 2012.
2. Tanenbaum A. S., Modern Operating Systems, 3/e, Pearson Education, 2008

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Discuss operating system concepts and analyze CPU scheduling algorithms. (PO- 1,2,3) (PSO-1,2,3)
2. Describe the various methods for accessing shared data and handling deadlocks. (PO-1,2,3) (PSO-1,2,3)
3. Employ the techniques for memory management during process execution and describe the file system and secondary storage structure. (PO-1,2,3) (PSO-1,2,3)
4. Describe the file system and secondary storage structure. (PO-1,2,3) (PSO-1,2,3)
5. Articulate the need for information protection mechanisms in OS and understand the working of modern operating system. (PO –1,2) (PSO-1,2,3)

OPERATIONS RESEARCH

Course Code: IS43 **Credits: 3:1:0**
Prerequisite: Nil **Contact Hours: 42L+14T**
Course Coordinator: Mrs. Shruthi G

Course Content

UNIT-I

Introduction to Operations Research (OR) and Linear Programming (LP): OR Models, Solving the OR Model, Phases of an OR Study; Modeling with Linear Programming (LP), Two-variable LP Model, Graphical LP solution, Solution of a Maximization/Minimization Model; Computer solution with Excel Solver; LP Model in Equation Form, Transition from Graphical to Algebraic Solution; TORA Tool

UNIT-II

The Simplex Method: The Simplex Method, Special cases in the Simplex Method, Degeneracy, Alternative optima, Unbounded solutions, Non-existing solutions; Artificial Variable Techniques: Two Phase Method, Big-M method; TORA Tool

UNIT-III

Duality and Game Theory: Introduction: Concept of Duality; Definition of Primal Dual Problems; General Rules for Converting any Primal into its Dual; Introduction to Dual Simplex Method; Computational Procedure of Dual Simplex Method; Illustrative Examples. Introduction of Game Theory, Characteristics of Games Theory; Minimax (Maximin) Criterion and Optimal Strategy; Saddle Point, Optimal Strategies and the value of game; Solution of Games with Saddle Point(s); Rectangular Games without Saddle Point; Arithmetic Method for (2x2) Games, Gambit tool for Game theory.

UNIT-IV

Project Management by PERT-CPM: Introduction, Applications of PERT/CPM Techniques, Basic Steps in PERT/CPM Techniques; Network Diagram representation, Rules for Drawing Network Diagram, Labeling Fulkerson‟s ‘I-J’ Rule; Time Estimates and Critical Path in Network Analysis; Project Evaluation and Review Technique; TORA tool.

UNIT-V

Transportation and Assignment Problems: Mathematical Formulation of transportation problem (TP); Matrix Form of TP; Feasible Solution, Basic Feasible Solution and Optimum Solution; Tabular Representation; Special Structure of Transportation Table and their Problems; Initial Basic Feasible Solution to TP; Moving Towards Optimality; Degeneracy in TP; Unbalanced TP; Mathematical Formulation of Assignment Problem (AP); Fundamental Theorems; Hungarian Method for AP; TORA tool.

Tutorial Exercises:

1. Problems on Formulation of Linear Programming Problems,
2. Problems on Graphical Solutions to Linear Programming Problems
3. Problems on Simplex Method
4. Problems on special cases in simplex method
5. Problems on Two Phase Method
6. Problems on Big-M Method
7. Problems on Dual Simplex Method
8. Problems on CPM
9. Problems on PERT
10. Problems on Game theory
11. Problems on Transportation Problems
12. Problems on Assignment Problems

Course Outcomes (COs):

At the end of the course, student will be able to -

1 Formulate and solve linear programming model using graphical method. (PO-1,2,3,5,12)(PSO-1,2)
2 Solve a linear programming model using simplex methods(PO-1,2,3,5)(PSO-1,2)
3 Apply game theory to model, analyze and solve real world problems. (PO-1,2,3,5,6,7)(PSO-1,2)
4 Analyze a Project network using PERT and CPM techniques.(PO-1,2,3,5,6,7)(PSO-1,2)
5 Solve Transportation and assignment problems. (PO-1,2,3,5,6,7)(PSO-1,2)

FINITE AUTOMATA AND FORMAL LANGUAGES

Course Code: IS44 **Credit: 3:1:0**
Prerequisite: Nil **Contact Hours: 42L+14T**
Course Coordinator: Dr. S R Mani Sekhar

Course Content

UNIT-I

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-II

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-III

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-IV

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-V

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:

- Problems on design of Deterministic Finite Automata.
- Problems on design of Non- Deterministic Finite Automata.
- Problems on Equivalence of DFA and NFA.
- Formulating Regular Expressions for given Regular Languages.
- Problems on Converting Finite Automata to Regular Expressions and vice versa.
- Problems on Minimization of DFA.
- Proving languages not to be regular using pumping lemma for regular languages.
- Problems on constructing Context free grammar given a context free language.
- Proving languages not to be context free using pumping lemma for context free languages.
- Problems on finding Ambiguity in grammars and Languages
- Problems on Simplification of Context-Free Grammars.

- Problems on design of Deterministic and Non-Deterministic Push Down Automata.
- Problems on Equivalence of CFG and PDA.
- Problems on design of Turing Machine.

Text Book:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson education, 2014.

Reference:

1. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Describe the basic concepts of automata theory and design the Finite State Models. ((PO-1, 2, 3) (PSO-1)
2. Formulate Regular Expression for Finite Automata and vice versa (PO-1, 2, 3) (PSO-1)
3. Design Context Free Grammar (CFG) in various Normal Forms and discuss their Applications. (PO-1, 2, 3) (PSO-1)
4. Design and Convert N-Push Down Automata and D-Push Down Automata for CFG and vice versa. (PO-1, 2, 3) (PSO-1)
5. Design Turing Machine for a given language and discuss its variants. ((PO-1, 2, 3) (PSO-1)

DESIGN AND ANALYSIS OF ALGORITHMS

Course Code: IS45 **Credit: 4:0:0**

Prerequisite: Fundamentals of Computing & Data Structures **Contact Hours: 56L**

Course Coordinator: Mrs. Deepthi. K

Course Content

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 Books:

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

Reference:

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

Course Outcomes (COs):

At the end of the course, students will be able to

1. Describe the fundamentals of algorithms, problem types and standard algorithm design techniques. (PO-1, 2,3) (PSO-1)
2. Develop algorithms using standard algorithm design techniques to solve computational problems. (PO-1, 2, 3,12) (PSO-1,2)
3. Analyze the efficiency of algorithms through mathematical models. (PO-1,2,3,12) (PSO-1,2)
4. Apply algorithms and design techniques to solve computational problems. (PO-1,2,3,126) (PSO-1,2)
5. Identify intractable problems and deal with the limitations of algorithm power. (PO-1,2,3) (PSO-1,2)

MICROCONTROLLERS

Course Code: IS46 **Credit: 4:0:0**
Prerequisite: Computer Organization and Architecture **Contact Hours: 56L**
Course Coordinator: Mr. Prashanth Kambli

Course content

UNIT-I

ARM Embedded Systems, RISC design philosophy, ARM design philosophy, Introduction to Embedded systems, Design of Embedded Systems, Applications, Embedded processors, Operating System, Connectivity.

Internal Components of System-on-chip, General Microprocessor UNIT-, MCU PIN diagram, Timers and Counters, Pulse width modulator.

UNIT-II

Serial Communication, Direct Memory Access, Semiconductor Memory, Designing Low power systems, BUS architecture.

Embedded Systems – the Software, Endian-ness, Data Alignment and Memory Banks, Peripheral I/O and Memory Mapped I/O, Load Store Architecture, Stack, FLAGS, Integrated Development Environment, Debugging.

UNIT-III

The Architecture of ARM 7, History of ARM, ARM 7 architecture, Interrupts and Exceptions, ARM7 pipeline, Advanced Features.

Assembly Programming of ARM7, Embedded program development, ARM7 Instruction set.

UNIT-IV

Assembly Language Programming, Accessing Memory, Programming of ARM7 using C**,** ARM7 SOC.

UNIT-V

Architecture of ARM Cortex-M, Cortex-M Processors, Cortex-M0, Modes and States, Programming Model, Memory Model, Nested Vector Interrupt Controller, Power management using sleep modes.

Text Books:

1. Andrew Sloss, Dominic Symes, Chris Wright, "ARM System Developer's Guide - Designing and Optimizing System Software", Elsevier Publication 2012.
2. Lyla B Das, "Architecture, Programming, and Interfacing of Low-power Procesors – ARM7, Cortex-M", Cenegage Learning India Pvt. Ltd. 2017.

Reference:

1. Kris Schindler, “Introduction to Microprocessor Based Systems using the ARM Processor”, Second Edition, Pearson Education, 2013

Course Outcomes (COs):

At the end of the course, students will be able to -

1. Make decisions about relevant embedded processor and choose an operating system and connectivity for a given application (PO-1,2)(PSO-1)
2. Describe the building blocks that contribute to the software aspects of embedded system design. (PO-1,2)(PSO-1)
3. Explain ARM7 architecture, interrupt structure and pipeline. (PO-1,2)(PSO-1,2)
4. Write ARM7 assembly code for a given problem (PO-1,2,3,4)(PSO-1,2)
5. Differentiate between versions of Cortex-M processors in terms of architecture, memory model and interrupt structure (PO-1,2,12)(PSO-3)

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

Course Code ISL47 **Credit: 0:0:1**

Prerequisite: Fundamentals of Computing & Data Structure **Contact Hours:14P**

Course Coordinator: Dr. Megha. P. Arakeri

Programs

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.

Text Books:

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

Reference:

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 (COs):

At the end of the course, students will be able to –

1. Implement the algorithms based on various design techniques. (PO-1,2,3,4,9,10) (PSO-1,2)
2. Analyze the efficiency of various algorithms. (PO-1,2,4,9,10)(PSO-1)
3. Produce substantial written documentation. (PO-9,10)(PSO-3)

MICROCONTROLLERS LABORATORY

Course Code ISL48 **Credit: 0:0:1**

Prerequisite: Nil **Contact Hours:14P**

Course Coordinator: Prashanth Kambli

Programs

1. **Familiarizing the Keil MicrovisionV IDE**
   a. Create a project, Edit an ASM file, Build, and Debug. Observe Disassembly window, Register and Memory contents in Step mode and in Run Mode.
   b. Execute a sample ARM Assembly Language Program to add two numbers in registers and store the sum in a register.
2. **ARM Assembly Language Programming Practice using Keil MicrovisionV # I**
   a. ALP to add first 5 natural numbers. Store sum in register.
   b. ALP to add first 10 odd numbers. Store sum in register.
   c. ALP to compute sum of 5 terms of an arithmetic progression. First term is 3, common difference is 7. Store sum in register.
   d. ALP to compute sum of squares of 5 numbers starting from 1. Write and use procedure SQU. Store sum in register.
3. **ARM Assembly Language Programming Practice using Keil MicrovisionV # II**
   a. ALP to add the first n even numbers. Store the result in a memory location.
   b. ALP to generate a geometric progression with a limit n. Display the results in memory.
4. **ARM ALP # I**
   a. ALP to find the arithmetic progression with a=3, d=7.
   b. ALP to find the sum of cubes of the first n natural numbers.
5. **ARM ALP # II**
   a. ALP to count the number of zeroes and ones in a binary number.
   b. ALP to find the average of ten 16-bit numbers stored in memory.
6. **ARM ALP # III**
   a. ALP to find the factorial of a number.
   b. ALP to generate the first n Fibonacci numbers.
7. **ARM ALP # IV**
   a. ALP to find the sum of digits of a number.
   b. ALP to convert BCD number to binary.
8. **ARM ALP # V**
   a. ALP to find nPr.
   b. ALP to find nCr.
9. **ARM ALP # VI**
   a. ALP to implement Bubble Sort on an array of integers.
   b. ALP to implement Binary Search on an array of integers.
10. **ARM ALP # VII**
    a. ALP to check whether the given number is palindrome.
    b. ALP to count the number of times a substring is repeated in the string.
11. **ARM C Programming *Practice using Keil MicrovisionV # I***
    a. C program to toggle the lowest pin of Port 0 with a delay between the two states. Observe and record the waveform obtained using the Logic Analyzer in the Keil simulator.
12. **ARM C Programming # I**
    a. C program to generate an asymmetric square wave of 120Hz and having a duty cycle of 25% using the Timer0 module.
    b. C program to generate a square wave using Timer0 in the interrupt mode.

13. **ARM C Programming # II**
    a. C program to make a LED glow at different brightness levels (low to high) with brightness levels varying over duration of 2s. Demonstrate using logic analyzer window.
14. **ARM C Programming # III**
    a. C program to display the string 'I LOVE ISE' in the serial window of UART1

Course Outcomes (COs):

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

1 Write assembly language programs for the ARM7 ISA. (PO-1,2,3) (PSO-1,2)
2 Write C programs for interfacing peripherals to the ARM7 MCU. (PO-1,2,3) (PSO-1,2)
3 Execute and Debug assembly language and C programs using a simulator. (PO-1,2,5) (PSO-1,2)

ADDITIONAL MATHEMATICS – II

Course Code: AM41 **Credit: 0:0:0**
Prerequisite: Nil **Contact Hours:40**
Course Coordinator: Dr. N. L. Ramesh

Course content:

UNIT-I

Differential calculus-II: Partial differentiation, Euler's theorem, total differential coefficient, differentiation of composite and implicit functions, Jacobian and Properties. Taylor's theorem for function of two variables, maxima and minima for functions of two variables.

UNIT-II

Vector integration: Line integrals, surface integrals and volume integrals. Green's theorem, Stokes' and Gauss divergence theorem (without proof) and problems, orthogonal curvilinear coordinates.

UNIT-III

Laplace transforms: Definitions, Laplace transforms of elementary functions, derivatives and integrals, periodic function, unit step function, inverse transforms, applications of Laplace transforms to solve differential equations.

UNIT-IV

Higher Order Differential Equations: Higher order linear differential equations, method of variation of parameters, Cauchy's and Legendre's homogeneous differential equations.

UNIT-V

Probability: Introduction. Sample space and events. Axioms of probability. Addition and multiplication theorems. Conditional probability-illustrative examples. Bayes theorem –examples.

Text Books:

1. B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers, 44th edition, 2017.
2. Erwin Kreyszig –Advanced Engineering Mathematics, Wiley publication, 10th edition, 2015.

Reference:

1. H.K. Dass – Higher Engineering Mathematics – S Chand Publications - 1998.
2. B.V. Ramana – Engineering Mathematics – Tata McGrawHill Publishing Co. Ltd. – New Delhi – 2008.

Course Outcomes (COs):

At the end of the course, students will be able to -

1. Find Jacobian, extreme values and power series expansion of a function. (PO-1, 2)
2. Exhibit the interdependence of line, surface and volume integrals using integral theorems. (PO-1, 2)
3. Use the concept of Laplace transforms to solve initial and boundary value problems (PO-1, 2)
4. Solve Linear differential equations with constant and variable coefficients (PO-1, 2)
5. Demonstrate the understanding of axioms and rules of probability to solve problems. (PO-1, 2)