Programme Outcomes – Computer Science and Engineering (UG)

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.

Programme Specific Outcomes

On completion of the B. Tech. (Computer Science & Engineering) degree the graduates will be able to

PSO1 - Pertain current knowledge and adapting to emerging applications of Mathematics, Science and Engineering fundamentals in the field of Computer Science and Engineering.

PSO2 - Apply standard Software Engineering practices and strategies in real-time software project development areas related to algorithms, networking, web design, cloud computing, IoT and data analytics of varying complexity using open-source programming environment or commercial environment to deliver quality product for the organization success.

PSO3 - Use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.

Programme Educational Outcomes – Computer Science and Engineering (UG)

PEO1: To prepare students to be ready for Industry/Research & Development/Academic profession so as to meet changing needs.

PEO2: To provide students with interdisciplinary skills necessary to formulate, analyze, and solve engineering challenges and equip the students with necessary skills for providing innovative solutions to real life problems.

PEO3: To absorb lifelong learning skills, managerial, leadership skills, and entrepreneurial ability in students and to inculcate in students; professional and ethical attitudes, communication and technical writing, team working attitude and an ability for relating engineering issues to social context.

Sharnbasva University, Kalaburagi
Scheme of Teaching and Examination 2018-19
Outcome Based Education(OBE) and Choice Based Credit System (CBCS)
(Effective from the academic year 2018-19)

III SEMESTER B. Tech.
(Common to Computer Science & Engineering and Information Science & Engineering)

Sl. No.	Course Code		Course Title	Teaching Dept. & Paper Setting Board	Teaching Hours/week			Examination				Credits
					L	T	P	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1	BSC	18MAT31	Engineering Mathematics-III	Mathematics	3			3	50	50	100	03
2	PCC	18CS/IS32	Data structures in C and Applications	CSE/ISE	3	1		3	50	50	100	04
3	PCC	18CS/IS33	Electronics Circuits and Logic Design	CSE/ISE	3	1		3	50	50	100	04
4	PCC	18CS/IS34	Computer Organization and Architecture	CSE/ISE	3	1		3	50	50	100	04
5	PCC	18CSL/ISL35	Data Structures Lab	CSE/ISE			2	3	50	50	100	01
6	PCC	18CSL/ISL36	Electronics Circuits and Logic Design Lab	CSE/ISE			2	3	50	50	100	01
7	PCC	18CSL/ISL37	UNIX Shell Programming Lab	CSE/ISE			2	3	50	50	100	01
8	PRJ	18CSP38/ISMP38	Project – III	CSE/ISE			2	3	50	50	100	01
9	HSMC	18HSMC39	YOGA	Humanities			2	2	50	50	100	01
Total					**12**	**3**	**10**	**26**	**450**	**450**	**900**	**20**

BSC-Basic Science, PCC-Professional Core, HSMC-Humanity and Social Science, PR-Project

Sharnbasva University, Kalaburagi
Scheme of Teaching and Examination 2018-19
Outcome Based Education(OBE) and Choice Based Credit System (CBCS)
(Effective from the academic year 2018-19)

IV SEMESTER B. Tech.
(Common to Computer Science & Engineering and Information Science & Engineering)

Sl. No.	Course Code		Course Title	Teaching Dept. & Paper Setting Board	Teaching Hours/week			Examination				Credits
					L	T	P	Duration in hours	CIE Marks	SEE Marks	Total Marks	
1	BSC	18MAT41	Engineering Mathematics-IV	Mathematics	3			3	50	50	100	03
2	PCC	18CS/IS42	Design and Analysis of Algorithms	CSE/ISE	3	1		3	50	50	100	04
3	PCC	18CS/IS43	Microprocessor	CSE/ISE	3	1		3	50	50	100	04
4	PCC	18CS/IS44	Java Programming	CSE/ISE	3	1		3	50	50	100	04
5	PCC	18CSL/IS45	Microprocessor Lab	CSE/ISE			2	3	50	50	100	01
6	PCC	18CSL/ISL46	Java Programming lab	CSE/ISE			2	3	50	50	100	01
7	PCC	18CSL/ISL47	Algorithm analysis and design Lab	CSE/ISE			2	3	50	50	100	01
8	PRJ	18CSP/ISP48	Project-IV	CSE/ISE			2	3	50	50	100	01
9	HSMC	18HSMC49	Official Kannada	Humanities			2	2	50	50	100	01
Total					**12**	**3**	**10**	**26**	**450**	**450**	**900**	**20**

BSC-Basic Science, PCC-Professional Core, HSMC-Humanity and Social Science, MP-Mini project

ENGINEERING MATHEMATICS-III

(Common to all branches)

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2019-20)

SEMESTER-III

Course Code :	**19MAT31**	**CIE Marks :**	**50**
Contact Hours/Week :	**04**	**SEE Marks:**	**50**
Total Hours:	**50**	**Exam Hours:**	**03**
Credits: 04			

Course Learning Objectives:

This course will enable students to:

- Introduce most commonly used analytical and numerical methods in the different engineering fields.
- Learn Laplace transform and Z-transforms, statistical methods, numerical methods.
- Solve the problem on Interpolation.
- To discuss the random variable and associated probability distributions.

Modules	**Teaching Hours**	**Revised Bloom's Taxonomy (RBT) Levels**
Module - 1		
LAPLACE TRANSFORMS : Definition, Transforms of Elementary functions, properties, periodic function, Unit step function, Unit impulse function. **INVERSE LAPLACE TRANSFORMS :** Definition, Convolution Theorem(without proof), Finding Inverse Laplace transform by convolution Theorem. Solution of Linear Differential equations using Laplace Transforms and Applications(5 Assignment Problem).	**10 - Hours**	**L1 & L2**
Module - 2		
Z- **TRANSFORMS:** Difference Equations ,Basic definitions, Damping rule, Shifting rule, Initial and Final Value theorems(without proof) and problems. Inverse Z-transforms. Applications of Z-transforms to solve difference equation(5 Assignment Problem).	**10 - Hours**	**L1 & L2**
Module - 3		
STATISTICAL METHODS: Correlation-karl Pearson's co-efficient of correlation problems.Regression analysis lines of regression (without proof)-problems. **CURVE FITTING:** Curve fitting by the method of least square. Fitting of the curves of the form $y = ax + b, y = ax^2 + bx + c$ & $y = ae^{bx}$. **Numerical Methods:** Numerical solution of algebraic and trans	**10 - Hours**	**L1 & L2**

cendental equations by Regula - Falsi Method and Newton-Raphson method. (5 Assignment Problem).		
Module - 4		
FINITE DIFFERENCE: Forward and Backward differences, Newton's forward and backward interpolation formulae.Divided difference-Newton's divided difference formulae.Lagrange's-interpolation formula and inverse interpolation formula(all formula without proof) problems. **NUMERICAL INTEGRATION:**Simpsons$(\frac{1}{3})^{rd}$, $(\frac{3}{8})^{th}$ rules, Weddle's rule (without proof) problems.(5 Assignment Problem).	**10 - Hours**	**L1 & L2**
Module - 5		
Probability Distribution: Random variables(discrete and continuous) probability mass/density functions. Binomial distribution, Poisson distribution. Exponential and Normal distributions. Problems.(5 Assignment Problem).	**10 - Hours**	**L3**
Course outcomes: On completion of this course, students are able to: • Know the use of Laplace transform and inverse Laplace transform in signal and image processing. • Explain the general linear system theory for continuous time signals and digital signal processing using the Z-transform. • Employ appropriate numerical methods to solve algebraic and transcendental equations. • Apply Green's Theorem, Divergence Theorem and Stokes' theorem in various application in the field of electro-magnetic and gravitational fields and fluid flow problems.		
Question paper pattern: • The question paper will have ten questions. • Each full Question consisting of 16 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.		
Text Books: *1. B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015.* *2. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons,10th Ed., 2015.*		
Reference Books: *1. N.P.Bali and Manish Goyal: A Text Book of Engineering Mathematics, Laxmi Publishers , 7th Ed., 2010.* *2. B.V.Ramana: "Higher Engineering Mathematics" Tata McGraw-Hill, 2006.* *3. H. K. Dass and Er. Rajnish Verma: "Higher Engineering Mathematics", S. Chand publishing, 1st edition, 2011.*		
Web Link and Video Lectures: *1. http://nptel.ac.in/courses.php?disciplineID=111* *2. http://www.khanacademy.org/* *3. http://www.class-central.com/subject/math*		

DATA STRUCTURES IN C AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – III

Subject Code	18CS/IS32	**CIE Marks**	50
Number of Lecture Hours/Week	03	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 04

Course objectives: This course will enable students

- To impart the basic concepts of data structures and algorithms.
- To understand concepts about searching and sorting techniques
- To understand basic concepts about stacks, queues, lists, trees and graphs.
- To enable them to write algorithms for solving problems with the help of fundamental data structures

Module I	**Teaching Hours**
Introduction: Data Structures, Data structure Operations, Pointers and Dynamic Memory Allocation, Data Abstraction. **Arrays and structures** :Arrays, dynamic allocated arrays ,structures & unions, Polynomials and Sparse Matrices. **Array Operations:** Traversing, inserting, deleting, searching, and sorting. **Strings**: Basic Terminology, Storing, Programming Examples. **RBT: L1, L2, L3.**	10Hours
Module II	
Stacks & Queues : **Stacks**: Definition, Stack Operations, Array Representation of Stacks, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression, **Recursion**: Factorial, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues:** Definition, Representation-array & linked representation of queues. Queue Operations, Circular Queues, Dequeues, Priority Queues. **RBT: L1, L2, L3.**	10 Hours
Module III	
Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Additional list operations-inverting singly linked list, concatenating singly linked list. Sparse matrix representation. **RBT: L1, L2, L3.**	08 Hours
Module IV	
Trees: Definition, Representation of trees, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations-copying binary tree, testing equality. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching. **RBT: L1, L2, L3.**	10 Hours
Module V	
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. **Sorting and Searching:** Insertion Sort, Radix sort, selection sort. **Hashing**: Hash Table organizations, Hashing Functions. **RBT: L1, L2, L3.**	10 Hours

Course Outcomes

After studying this course, students will be able to:

CO 1: For a given Search problem (Linear Search and Binary Search) student will able to implement it.

CO 2: For a given problem of Stacks, Queues and linked lists student will be able to implement its applications.

CO 3: Write functions on different types of trees and their operations.

CO 4: To implement Graphs, Searching, Sorting, Hashing and their applications.

CO5: Implement Hashing and their Applications.

.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Fundamentals of Data Structures in C - Ellis Horowitz and Sartaj Sahni, 2nd edition, Universities Press, 2014.
2. Data Structures - Seymour Lipschutz, Schaum's Outlines, Revised 1st edition, McGraw Hill, 2014

Reference Books:

1. Data Structures: A Pseudo-code approach with C –Gilberg & Forouzan, 2nd edition, Cengage Learning, 2014.
2. Data Structures using C, , Reema Thareja, 3rd edition Oxford press, 2012.
3. An Introduction to Data Structures with Applications- Jean-Paul Tremblay & Paul G. Sorenson, 2nd Edition, McGraw Hill, 2013.
4. Data Structures using C - A M Tenenbaum, PHI, 1989.
5. Data Structures and Program Design in C - Robert Kruse, 2nd edition, PHI, 1996.

ELECTRONICS CIRCUITS AND LOGIC DESIGN

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – III

Subject Code	18CS/IS33	**CIE Marks**	50
Number of Lecture Hours/Week	03	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS - 03

Course objectives: This course will enable students to

- Recall and Recognize construction and characteristics of JFETs and MOSFETs and differentiate with BJT
- Demonstrate and Analyze Operational Amplifier circuits and their applications
- Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques.
- Describe and Design Decoders, Encoders, Digital multiplexers, Adders and Subtractors, Binary comparators, Latches and Master-Slave Flip-Flops.
- Describe, Design and Analyze Synchronous and Asynchronous Sequential.
- Explain and design registers and Counters, A/D and D/A converters.

Module I	**Teaching Hours**
Field Effect Transistors: Junction Field Effect Transistors, MOSFETs, Differences between JFETs and MOSFETs, Biasing MOSFETs, FET Applications, CMOS Devices. Wave-Shaping Circuits: Integrated Circuit(IC) Multivibrators. Introduction to Operational Amplifier: Ideal v/s practical Opamp, Performance Parameters,. **RBT: L1, L2.**	10 Hours
Module II	
The Basic Gates: Review of Basic Logic gates, positive and negative logic Introduction to HDL. Combinational Logic Circuits: Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs, Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by QuineMcClusky Method, Hazards and Hazard covers, HDL Implementation Models. **RBT: L1, L2.**	10 Hours
Module III	
Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit Flip- Flops: RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIPFLOPs. **RBT: L1, L2.**	08 Hours
Module IV	
Flip- Flops: FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, HDL Implementation of FLIP-FLOP. Registers: Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register implementation in HDL. Counters: Asynchronous Counters, Decoding Gates,3s. **RBT: L1, L2.**	10 Hours

Module V	
Design of synchronous and asynchronous sequential circuits: model selection, state transition diagram, state synthesis table design equation and circuit diagram, implementation using read only memory. **D/A Conversion and A/D Conversion:** Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, Continuous A/D Conversion, A/D Techniques, Dual-slope A/D Conversion . **RBT: L1, L2.**	10Hours

Course Outcomes

After Studying this course, students will be able to

CO 1: Acquire knowledge of

- JFETs and MOSFETs , Operational Amplifier circuits and their applications
- Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique.
- Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors.

CO 2: Working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters.

CO 3: Analyze the performance of

- JFETs and MOSFETs , Operational Amplifier circuits
- Simplification Techniques using Karnaugh Maps, Quine McClusky Technique.
- Synchronous and Asynchronous Sequential Circuits.

CO 4: Apply the knowledge gained in the design of Counters, Registers and A/D & D/A converters.

CO 5 : To Design Synchronous and Asynchronous Sequential Circuits.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, 2012.
2. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015

Reference Books:

1. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design with VHDL, 2nd Edition,Tata McGraw Hill, 2005.
2. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine-Pearson, 2010.
3. M Morris Mano: Digital Logic and Computer Design, 10 th Edition, Pearson, 2008

Computer Organization & Architecture

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – III

Subject Code	18CS/IS34	**CIE Marks**	50
Number of Lecture Hours/Week	03	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 03

Course objectives: This course will enable students

- How Computer Systems work & the basic principles
- Instruction Level Architecture and Instruction Execution.
- The current state of art in memory system design
- How I/O devices are accessed and its principles.
  To provide the knowledge on Instruction Level Parallelism.
- To impart the knowledge on micro programming.
- Understand Concepts of advanced pipelining techniques, Computer Arithmetic and parallel processing

Module I	**Teaching Hours**
Functional blocks of a computer: Functional units, Basic operational concepts, Bus Structure, Software, and Performance. Signed number representation, character representation. Memory location and address, Instruction and sequencing, Basic IO operations, Addressing Modes, Additional Instructions: Shift and Rotate Instructions **RBT: L1, L2, L3.**	10Hours
Module II	
Basic Processing Unit: Single Bus Organization, Multiple Bus Organization, Hardwired and micro-programmed design approaches. **Input Output Organization**: Accessing I/O devices, Interrupts, DMA, Buses. **RBT: L1, L2, L3.**	10 Hours
Module III	
The Memory System: Semiconductor RAM memories (SDRAM, ADRAM), Cache Memories, Performance Consideration. **Arithmetic:** Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication. **RBT: L1, L2, L3.**	10 Hours
Module IV	
Pipelining: Introduction, Major Hurdles of Pipelining, How is pipelining implemented?, What makes pipeline hard to implement, Instruction Level Parallelism: Concepts and Challenges. **RBT: L1, L2, L3.**	10 Hours
Module V	
Memory Hierarchy: Introduction, Cache Performance, Six basic Cache Optimization, Virtual Memory, Memory Hierarchy Design: 10 Advanced optimizations of cache performances. **RBT: L1, L2, L3.**	08 Hours

Course Outcomes

After studying this course, students will be able to:
CO1: Identify basic structure of computer and its performance measures.

CO2: Demonstrate functioning of different sub systems, such as processor, Input/output, and memory.
CO3: Understand the concepts of Pipelining and parallel processor.
CO4: Understand the concepts of Memory organization.
CO5: Understand the concepts of, Arithmetic Operations and Characters.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Carl Hamacher, Z. Vranesic & S.Zaky, "Computer Organization",5th Edition , Tata McGraw-Hill Publishing Company Ltd. New Delhi, 2002.
2. John L. Hennessy and David A. Patterson, Computer Architecture: A quantitative approach, 5th edition, Morgan Kaufmann Elseveir, 2013

Reference Books:

1. Morris Mano, "Computer System Architecture", PHI, 19862.William Stallings Computer Organization & Architecture, 7th Edition , PHI 2006.
2. Kai Hwang and Naresh Jotwani, Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability, McGraw Hill Education 3/e. 2015.

Data Structures Lab

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – III

Subject Code	18CSL/ISL35	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students

- To design, develop, test and debug in C/C++ language considering appropriate data structure.
- Illustrate and implement data types such as stack, queue and linked list and apply them for the given problem.
- Illustrate and implement the trees and other data structures.

PART-A

Students are required to implement following programs using C/C++.

1. Implementation of stack ADT using arrays
2. Implementation of queue ADT using arrays
3. Implementation of List ADT
4. Implementation of Graph ADT using List
5. Implementation of tree ADT using List / Array

Part B

Application of Stack

1. Implementation of Infix to Postfix conversion.
2. Implementation of postfix evaluation.

Application of Queue

3. Implementation of Priority queue program using array.
4. Implementation of multiple stacks and queues

Application of List

5. Implementation of sparse matrix multiplication.
6. Implementation of Linked Lists menu driven program (stack and queue)

Application of Graph & Tree

7. Implementation of construction of expression tree using postfix expression.
8. Implementation of various operations on tree like – copying tree, counting the number of nodes in the tree.
9. Implementation of Binary Heap program

Course Outcomes

After studying this course, students will be able to:

CO 1: Understand and explore the fundamental concepts of various data structures.
CO 2: Analyze and represent various data structures
CO 3: Design algorithms on different data structures like Stack, Queue, List, Tree and hashing.
CO 4: Implement programs with suitable data structure based on the requirements of the application

Electronics Circuits & Logic Design Lab

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – III

Subject Code	18CSL/ISL36	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students

PART A

1. A. Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.
   B. Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.

2. A. Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.
   B. Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.

3. Design and implement an A stable multi vibrator circuit using 555 timer for a given frequency and duty cycle.

PART B

1. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.

2. Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
3. Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify it's working.
4. Design and implement code converter
   I)Binary to Gray
   II) Gray to Binary Code using basic gates.

5. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.
6. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
   b) Design and develop the Verilog / VHDL code for D Flip-Flop with positive edge triggering.

Course Outcomes

After studying this course, students will be able to:
CO 1: Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
CO2: Design and demonstrate various combinational logic circuits.
CO3: Design and demonstrate various types of counters and Registers using Flip-flops
CO4: Use simulation package to design circuits.
CO5: Understand the working and implementation of ALU.

Unix Shell Programming Lab

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – III

Subject Code	18CSL/ISL37	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students

- To Study of UNIX basic Commands
- To introduce Basic Unix general purpose Commands.
- To write shell scripts to solve problems

Part A

1. Study of UNIX basic commands:
   cal, date, echo, printf, bc, script, mailx, passwd, who, uname, tty, stty, pwd, cd, mkdir, rmdir, ls, cat, cp, rm, mv, more, file, wc, od, cmp,comm, diff, chmod.
2. Study of vi editor.
3. Write a script to study if...else, if and case statements.
4. Write a script to study for, while and until.
5. Study the Filters for stream handling features of the shell for input and output.
   E.g. pr, head, tail, cut, paste, sort, nl, uniq, tr.

Part B

1.
   a) Write a Shell program to count number of user's login and print first login user information
   b) Write Shell Script to read user name and find whether the user is currently working in the system or not.
2.
   a) Write shell script for-
      (i) Showing the count of users logged in.
      (ii) Printing Column list of files in your home directory.
      (iii) Listing your job with below normal priority.
      (IV) Continue running your job after logging out.
   b) Write a shell script to create a file. Follow the instructions
      (i) Input a page profile to yourself, copy it into other existing file;
      (ii) Start printing file at certain line.
      (iii) Print all the difference between two file, copy the two files.
      (iv) Print lines matching certain word pattern.
3.
   a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
   b) Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
4.
   a) Write a shell script that receives any number of file names as arguments checks if every argument is a file or directory, when it is a file, report no of lines in it.
   b) Write a shell script that accepts a list of file names as its arguments, count and reports the

occurrence of each word that is present in the first argument file on other argument files.

5.
   a) Write a grep/egrep script to find the number of words character, words and lines in a file.
   b) Write an egrep script to display list of files in the directory.

6.
   a) Write an awk script to count the number of lines in a file that do not Contain vowels.
   b) Write an awk script to find the number of characters, words and lines in a file.

7.
   a) Write a Perl script to compute the power of a given number.
   b) Write a Perl script to check a number is prime or not.

Course Outcomes

After studying this course, students will be able to:
CO1: Work on any Unix platform with confidence
CO2: Write effective scripts for their day to day jobs
CO3: Understand and use most of the Unix features and commands
CO4: Will be able to do Basic System administration.

ADDITIONAL MATHEMATICS - I

(B.Tech. III semester Common to all branches)

(A Bridge course for Lateral Entry students of III Sem. B. Tech.)

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2019-20)

Course Code : 19MATDIP31	**CIE Marks : 00**
Contact Hours/Week : 03	**SEE Marks: 100**
Total Hours:40	**Exam Hours:03**
Semester : III	**Credits: 00**

Course Learning Objectives:

This course will enable students to:

- Acquire basic concepts of complex trigonometry, vector algebra, differential & integral calculus and vector differentiation.
- Evaluation of double and triple integrals.
- know the basic concepts of partial differential equations.
- To develop the knowledge of matrices and linear algebra in compressive manner.
- To understand the essential concept of linear algebra.

MODULE-I

Complex Trigonometry-1:

Complex Numbers: Definition and Properties . Modulus and Amplitude of complex number, Argand's diagram , De-Moivre's theorem (without proof)

Vector Analysis : Scalar and Vectors. Vector addition and subtraction. Multiplication of vectors (Dot and Cross products) Scalar and vector triple products- simple problems, Vector Differentiation : Gradient, Divergence and Curl.

8 - Hours

MODULE-II

Differential Calculus:

Review of successive differentiation. Formulae of N^{th} derivatives of standard functions- Leibnitz's theorem (without proof).

Polar Curves: Expression for Angle between radius vector and tangent, length of perpendicular from pole to the tangent, angle between two polar curves, Pedal Equation of polar curves and problems. Taylor' and Maclaurin's seires expansions.

8 - Hours

MODULE-III

Partial Differentiation :

Definitions of Partial Differentiation, Direct and Indirect partial derivatives, Symmetric functions, Homogeneous function and Euler's theorem on homogeneous function. Total Derivative of composite and implicit function. Jacobian.

8 - Hours

MODULE-IV

Integral Calculus:

Reduction Formulae of $\int_0^{\pi/2} Sin^n x dx$, $\int_0^{\pi/2} Cos^n x dx$, and Statement of Reduction formulae $\int_0^{\pi/2} Sin^m x Cos^n x dx$ and Problems.

Double and Triple integral- simple problems.

8 - Hours

MODULE-V

Linear Algebra:

Basic concepts of matrices- Rank of matrix by elementary row transformations- Echelon form. Consistency of system of Linear equations. Solution of system linear equations by Gauss Elimination method, Linear Transformation, Cayley- Hamilton theorem to compute inverse of matrix. Eigen values and Eigen vector, Largest eigen values of eigen vectors by Reyleigh's Power method. **8 - Hours**

Course outcomes:

On completion of the course, students are able to:

- Understand the fundamental concepts of complex numbers and vector algebra to analyze the problems arising in related area.
- Use derivatives and partial derivatives to calculate rates of change of multivariate functions.
- Learn techniques of integration including double and triple integrals to find area, volume, mass and moment of inertia of plane and solid region.
- Analyze position, velocity and acceleration in two or three dimensions using the calculus of vector valued functions.
- Recognize and solve first-order ordinary differential equations occurring in different branches of engineering.
- Solve systems of linear equations in the different areas of linear algebra.

Question paper pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Book:

B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, New Delhi, 43rd Ed., 2015.

Reference Books:

1. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed., 2015.

2. N.P.Bali and Manish Goyal: Engineering Mathematics, Laxmi Publishers, 7th Ed., 2007.

ENGINEERING MATHEMATICS-IV

(Common to all branches)

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2019-20)

SEMESTER-IV

Course Code :	**19MAT41**	**CIE Marks :**	**50**
Contact Hours/Week :	**04**	**SEE Marks:**	**50**
Total Hours:	**50**	**Exam Hours:**	**03**

Credits: 04

Course Learning Objectives:

This course will enable students to:

- Learn Fourier series and Fourier transforms.
- Conversant with numerical methods to solve ordinary differential equations, complex analysis, joint probability distribution and stochastic processes arising in science and engineering.

Modules	**Teaching Hours**	**Revised Bloom's Taxonomy (RBT) Levels**
Module - 1		
Fourier Series: Periodic functions, Dirichlet's condition, Fourier Series of periodicfunctions with period 2π and with arbitrary period 2c. Fourier series of even and odd functions Half range Fourier Series, practical harmonic analysis(5 Assignment Problem).	**10 - Hours**	**L1 & L2**
Module - 2		
Fourier Transforms: Infinite Fourier transforms, Fourier sine and cosine transforms. Inverse Fourier-transform (5 Assignment Problem). **Complex line Integrals:** Cauchy's Integration theorem, Cauchy integral formula, Laurent's Series, types of singularities. Residue, Poles, Cauchy's Residue theorem (without proof) and Problems. **Transformations:** Bilinear transformations and problems.	**10 - Hours**	**L1 & L2**
Module - 3		
Numerical Methods: Numerical solution of ordinary differential equations of first order and first degree, Taylor's series method, modified Euler's-method Runge Kutta method of fourth order. Milne's and Adams- Bashforth predictor and corrector methods (No derivations of formulae). (5 Assignment Problem).	**10 - Hours**	**L1 & L2**
Module - 4		
Numerical Methods: Numerical solution of second order ordinary differential equations, Runge- Kutta Method and Milne's Method, Numerical solution of P.D.E: Numerical solution of heat equation, wave equation , problems. (5 Assignment Problem).	**10 - Hours**	**L1 & L2**

Module - 5		
Joint probability distribution: Joint Probability distribution for two discrete random variables, expectation, covariance, correlation coefficient **Stochastic process:** Stochastic processes, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains, higher transition probability- simple problems.(5 Assignment Problem).	**10 - Hours**	**L3**

Course Outcomes: On completion of this course, students are able to:

- Know the use of periodic signals and Fourier series to analyze circuits and system communications.
- Explain the general linear system theory for continuous time signals and digital signal processing using the Fourier Transform.
- Solve first and second order ordinary differential equations arising in flow problems using single step and multistep numerical methods.
- Understand the analyticity, potential fields, residues and poles of complex potentials in field theory and electromagnetic theory.
- Describe bilinear transformation arising in aerofoil theory, fluid flow visualization and image processing.
- Solve problems on probability distributions relating to digital signal processing, information theory and optimization concepts of stability of design and structural engineering.
- Determine joint probability distributions and stochastic matrix connected with the multivariable correlation problems for feasible random events.
- Define transition probability matrix of a Markov chain and solve problems related to discrete parameter random process.

Question paper pattern:

- The question paper will have ten questions.
- Each full Question consisting of 16 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015.
2. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons,10th Ed., 2015.

Reference Books:

1. N.P.Bali and Manish Goyal: A Text Book of Engineering Mathematics, Laxmi Publishers , 7th Ed., 2010.
2. B.V.Ramana: "Higher Engineering Mathematics" Tata McGraw-Hill, 2006.
3. H. K. Dass and Er. Rajnish Verma: "Higher Engineering Mathematics", S. Chand publishing, 1st edition, 2011.

Web Link and Video Lectures:

1. *http://nptel.ac.in/courses.php?disciplineID=111*
2. *http://www.khanacademy.org/*
3. *http://www.class-central.com/subject/math*

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – IV

Subject Code	18CS/IS42	**CIE Marks**	50
Number of Lecture Hours/Week	03	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 03

Course objectives: This course will enable students to

- Explain various computational problem solving techniques.
- Apply appropriate method to solve a given problem.
- Describe various methods of algorithm analysis.

Module I	**Teaching Hours**
Introduction: Notion of Algorithm, Review of Asymptotic Notations, Mathematical analysis of Non-Recursive and recursive Algorithms with Examples. Important Problem Types: Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. Fundamental Data Structures: Stacks, Queues, Graphs, Trees, Sets and Dictionaries. **RBT: L1, L2, L3.**	10Hours
Module II	
Divide and Conquer: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum , Merge sort, Quick sort , Advantages and Disadvantages of divide and conquer. Decrease and Conquer Approach: Topological Sorting. **RBT: L1, L2, L3.**	10 Hours
Module III	
Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines. Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm . Single source shortest paths: Dijkstra's Algorithm . Optimal Tree problem: Huffman Trees and Codes . Transform and Conquer Approach: Heaps and Heap Sort. **RBT: L1, L2, L3.**	08 Hours
Module IV	
Dynamic Programming: General method with Examples. Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Knapsack problem,Bellman-Ford Algorithm, Travelling Sales Person problem. **RBT: L1, L2, L3.**	10 Hours
Module V	
Backtracking: General method , N-Queens problem , Sum of subsets problem , Graph coloring , Hamiltonian cycles . Branch and Bound: Assignment Problem, Travelling Sales Person problem , 0/1 Knapsack problem : LC Branch and Bound solution , FIFO Branch and Bound solution. **RBT: L1, L2, L3.**	10 Hours

Course Outcomes

After studying this course, students will be able to:

CO 1: Describe computational solution to well known problems like searching, sorting etc.
CO 2: Estimate the computational complexity of different algorithms.
CO 3: Devise an algorithm using appropriate design strategies for problem solving.
CO 4: Describe the General Methods and implement the programs by using various Dynamic Programming algorithms.

CO 5: Understand the Backtracking algorithms and implement the programs.
Question paper pattern: • The question paper will have ten questions. • There will be 2 questions from each module. • Each question will have questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.
Text Books: 1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2nd Edition, 2009. Pearson. 2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press/
Reference Books: 1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI. 2. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education).

MICROPROCESSOR & MICROCONTROLLER

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – IV

Subject Code	18CS/IS43	**CIE Marks**	50
Number of Lecture Hours/Week	03	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 03

Course objectives: This course will enable students to

- To impart basic understanding of the internal organization of 8086/88 Microprocessor.
- To introduce the concepts of interfacing microprocessors with external devices.
- To develop Assembly language programming skills.

Module I	**Teaching Hours**
The x86 microprocessor: Brief history of the x86 family, Inside the 8088/86,Introduction to assembly programming, Introduction to Program Segments, The Stack, Flag register, x86 Addressing Modes. **Assembly language programming:** Directives & a Sample Program, Assemble, Link & Run a program, More Sample programs, Control Transfer Instructions, Data Types and Data Definition, **RBT: L1, L2.**	10Hours
Module II	
x86: Instructions sets description**, Arithmetic and logic instructions and programs:** Unsigned Addition and Subtraction, Unsigned Multiplication and Division, Logic Instructions,BCD and ASCII conversion, Rotate Instructions. **INT 21H and INT 10H Programming:** Bios INT 10H Programming , DOS Interrupt 21H. 8088/86 Interrupts,x86 PC and Interrupt Assignment. **RBT: L1, L2.**	10 Hours
Module III	
Signed Numbers and Strings: Signed number Arithmetic Operations, String operations, **Memory and Memory interfacing: M**emory address decoding, data integrity in RAM and ROM, 16-bit memory interfacing. **8255 I/O programming:** I/O addresses MAP of x86 PC's, programming and interfacing the 8255. **RBT: L1, L2.**	08 Hours
Module IV	
Microprocessors versus Microcontrollers, **ARM Embedded Systems :**The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software, **ARM Processor Fundamentals :** Registers , Current Program Status Register , Pipeline, Exceptions, Interrupts, and the Vector Table. **RBT: L1, L2.**	10 Hours
Module V	
Introduction to the ARM Instruction Set : Data Processing Instructions ,Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions Coprocessor Instructions, Loading Constants. **RBT: L1, L2.**	10 Hours
Course Outcomes	

After studying this course, students will be able to:

CO 1: Differentiate between microprocessors and microcontrollers

CO 2: Develop assembly language code to solve problems

CO 3: Interface microprocessors with various external devices.

CO 4: Analyze and compare the features of microprocessors.

CO5: Demonstrate interrupt routines for interfacing devices.

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Danny Causey, The x86 PC Assembly Language Design and Interfacing, 5th Edition, Pearson, 2013.
2. **ARM system developers guide**, Andrew N Sloss, Dominic Symes and Chris Wright, Elsevier, Morgan Kaufman publishers, 2008.

Reference Books:

1. Douglas V. Hall: Microprocessors and Interfacing, Revised 2nd Edition, TMH, 2006.
2. Ayala : The 8086 Microprocessor: programming and interfacing - 1st edition, Cengage Learning
3. The Definitive Guide to the ARM Cortex-M3, by Joseph Yiu, 2nd Edition , Newnes, 2009
4. The Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005
5. ARM System-on-Chip Architecture, Steve Furber, Second Edition, Pearson, 2015
6. Architecture, Programming and Interfacing of Low power Processors- ARM7, Cortex-M and MSP430, Lyla B Das Cengage Learning, 1st Edition.

JAVA PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)
SEMESTER – IV

Subject Code	18CS/IS44	**CIE Marks**	50
Number of Lecture Hours/Week	03	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 0

Course objectives: This course will enable students to

- Learn fundamental features of object oriented language and JAVA
- Set up Java JDK environment to create, debug and run simple Java programs.
- Learn object oriented concepts using programming examples.
- Study the concepts of importing of packages and exception handling mechanism.
- Discuss the String Handling examples with Object Oriented concepts.

Module I	**Teaching Hours**
An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings **RBT: L1, L2.**	10Hours
Module II	
Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java's Selection Statements, Iteration Statements, Jump Statements. Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, A Stack Class. **RBT: L1, L2.**	10 Hours
Module III	
A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class. **RBT: L1, L2, L3.**	10 Hours
Module IV	
Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions. **RBT: L1, L2, L3.**	08 Hours

Module V	
The Applet Class: Introduction, Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting; Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase(); ApletContext and showDocument(); The AudioClip Interface; The AppletStub Interface; Output to the Console. String Handling: The String Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using valueOf( ), Changing the Case of Characters Within a String , Additional String Methods, StringBuffer, StringBuilder. **RBT: L1, L2, L3.**	10 Hours

Course Outcomes

The students shall able to:
CO 1: Explain the object-oriented concepts and JAVA.
CO 2: Develop computer programs to solve real world problems in Java.
CO 3: Develop the computer programs using exception concepts
CO 4: Understand the concepts of inheritance.
CO 5: Develop simple GUI interfaces for a computer program to interact with users

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:
1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007. (Chapters 2, 3, 4, 5, 6,7, 8, 9,10, 12,13,15)

Reference Books:
1. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education,2008, ISBN:9788131720806.
2.RajkumarBuyya,SThamarasiselvi, xingchenchu, Object oriented Programming with java, Tata McGraw Hill education private limited.
3. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies.
4. Anita Seth and B L Juneja, JAVA One step Ahead, Oxford University Press, 2017.

MICROPROCESSOR AND MICROCONTROLLER LAB

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – IV

Subject Code	18CSL/ISL45	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students to

- Demonstration and Explanation of hardware components ,8086 architecture, pin diagram
- Develop and execute the following programs using 8086 Assembly Language. Any suitable assembler like MASM/TASM/8086 kit or any equivalent software may be used.

Laboratory Session-1: Write-up on Microprocessors, 8086 Functional block diagram, Pin diagram and description. The same information is also taught in theory class; this helps the students to understand better

Laboratory Session-2: Write-up on Instruction group, Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 20 marks as lab experiments.

PART – A

1. Design and develop an assembly language program to search a key element "X" in a list of 'n' 16-bit numbers. Adopt Binary search algorithm in your program for searching.
2. Design and develop an assembly program to sort a given set of 'n' 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.
3. Design and develop an assembly language program to read the current time and Date from the system and display it in the standard format on the screen.
4. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.
5. Design an assembly language program to compute nCr using recursive procedure. Assume that 'n' and 'r' are non-negative integers.
6. Design an assembly language program to create a file and delete an existing file.
7. To write and simulate C Program to ARM microprocessor using KEIL. (Demonstrate with the help of suitable program)

PART – B

1. Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.
2. Design and develop BCD Up-Down counter using Logic Controller Interface.
3. Design and develop an assembly program to display messages "FIRE" and "HELP" alternately with flickering effects on a 7-segment display interface for a suitable period of time.
4. To interface stepper motor with ARM processor- ARM7TDMI/LPC2148. Write a program to rotate stepper motor.

Course Outcomes

The students should be able to:
CO 1: Program a microprocessor to perform arithmetic, logical and data transfer applications.
CO 2: Understand assembler directives, DOS Interrupts, branch and loop operations.

CO 3: Interface a microprocessor to various devices for simple applications. CO 4: Effectively utilize microprocessor peripherals. CO 5: Utilize procedures and macros for modular programming

JAVA PROGRAMMING LAB

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – IV

Subject Code	18CSL/ISL46	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students

- Learn fundamental features of object oriented language and JAVA
- Set up Java JDK environment to create, debug and run simple Java programs.
  Learn object oriented concepts using programming examples

PART – A

1. a. Write a JAVA program to implement class mechanism. –Create a class, methods and invoke them inside main method.
   b. Write a JAVA program to implement shift operators in JAVA
2. a. Write a JAVA program to implement constructor overloading.
   b. Write a JAVA program to implement for-each loop to compute average of n natural numbers.
3. a. Write a JAVA program to implement multi level Inheritance.
   b. Write a JAVA program for abstract class to find areas of different shapes.
4. a. Write a JAVA program that describes exception handling mechanism.
   b. Write a JAVA program to implement break and continue statements.
5. a. Write a JAVA program using IO Streams.
   b. Write a JAVA program using files.

PART – B (Implement the following in JAVA)

1. Write a JAVA program that creates threads by extending Thread class .First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds,(Repeat the same by implementing Runnable.
2. Write a JAVA program Producer Consumer Problem.
3. Write a JAVA program to create an applet and set its background color and foreground color displaying a message
4. A. Write a JAVA program to demonstrate key event handlers using delegation event model.

The students should be able to:
CO 1: Implement the java program using constructor, inheritance.
CO2: Implement the java program using exception handling.
CO2: Implement the java program using threads.

DESIGN AND ANALYSIS OF ALGORITHM LAB

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2018-2019)
SEMESTER – IV

Subject Code	18CSL/ISL47	**CIE Marks**	50
Number of Lecture Hours/Week	02	**SEE Marks**	50
Total Number of Lecture Hours	48	**Exam Hours**	03

CREDITS – 01

Course objectives: This course will enable students

- Design and implement various algorithms in JAVA
- Employ various design strategies for problem solving.
- Measure and compare the performance of different algorithms.

PART – A

1. Design a program to search a key element of n integers using binary search algorithm and compute time complexity
2. Design a program to Sort a given set of n integer elements using Quick Sort method and compute its time complexity.
3. Design a program to sort set of n integer elements using Merge Sort method and compute its time complexity.
4. Implement the 0/1 Knapsack problem using
   (a) Dynamic Programming method.
   (b) Greedy method.
5. Design a program to print all the node reachable from a given starting node in a given digraph using DFS method.

PART – B (Implement the following in JAVA)

1. Write a Program find shortest paths to other vertices using Dijkstra's algorithm.
2. (a) Write a program to find a Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.
   (b) Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.
3. Write a program to
   (a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm.
   (b) Implement transitive closure using warshall Algorithm.
4. Design and implement to find a subset of a given set.
5. Implement Travelling Salesman problem using Dynamic program.

Course Outcomes

The students should be able to:
CO 1: Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
CO 2: Develop variety of algorithms such as sorting, graph related, combinatorial, etc., in a high level language.
CO 3: Analyze and compare the performance of algorithms using language features.
CO 4: Apply and implement learned algorithm design techniques and data structures to solve real-world problems. Will be able to do Basic System administration.

ADDITIONAL MATHEMATICS - II

(B.Tech. III semester Common to all branches)

(A Bridge course for Lateral Entry students of IV Sem. B.Tech.)

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2019-20)

Course Code : 19MATDIP41	**CIE Marks : 00**
Contact Hours/Week : 03	**SEE Marks: 100**
Total Hours:40	**Exam Hours:03**
Semester : IV	**Credits: 00**

Course Learning Objectives:

This course will enable students to:

- Solve first order differential equations. .
- Solve second and higher order differential equations.
- Understand and solve the partial differential equation.
- To acquire the knowledge of elementary probability theory.
- Know the basic concepts of evaluation of double and triple integrals.

MODULE-I

Differential Equation-1:-

Solution of first order and first degree differential equations: Variable separable, Homogeneous, Exact and Reducible to exact differential equation, Linear differential equation. Applications of first order first degree differential equations: Newton's law of cooling.

8 - Hours

MODULE-II

Differential Equations-2:-Solution of second & higher order Ordinary linear differential equation with constant co-efficients.Method of variation of parameters. Solution of homogeneous LDE by Power series solution Method.

8 - Hours

MODULE-III

Partial Differential Equations(PDE's):- Formation of PDE by eliminating arbitrary constant & functions, Solution of Non-homogeneous PDE by direct integration, solution of homogeneous PDE with respect to one independent variable only. Derivation of one dimensional wave equation and heat equation and Various possible solution of wave & heat equations by methods of separation of variables.

8 - Hours

MODULE-IV

Improper Integrals: Beta and gamma functions and its properties and examples.

Evaluation of double integral over a specific region, changing the order of integration , changing into polar form.

8 - Hours

MODULE-V

Probability: Introduction , Sample space and Events. Axioms of Probability, Addition & Multiplication theorems. Conditional probability- illustrative examples. Baye's theorem-examples.

8 – Hours

Course Outcomes:

On completion of this course, students are able to:

- Solve first order differential equations in the different areas of Engineering.
- Solve second and higher order differential equations occurring in of electrical circuits, damped/un-damped vibrations.
- Solve second order partial differential equations in the different areas in the real world.
- Recall basic concepts of elementary probability theory and, solve problems related to the decision theory, synthesis and optimization of digital circuits.
- To find the surface area and volume of 3D objects.

Question paper pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four subquestions) from each module.
- Each full question will have sub questions covering all the topicsunder a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Book:

B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015.

Reference Books:

1. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed., 2015.

2. N.P.Bali and Manish Goyal: A Text Book of Engineering Mathematics, Laxmi Publishers,7th Ed., 2007.