

K.S.INSTITUTE OF TECHNOLOGY, BANGALORE

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM)

DEPARTMENT OF COMPUTER SCIENCE & ENGG.

B. E. COMMON TO ALL PROGRAMMES SEMESTER - III			
TRANSFORM CALCULUS, FOURIER SERIES AND NUMERICAL TECHNIQUES			
Course Code	18MAT31	CIE Marks	40
Teaching Hours/Week (L: T:P)	(2:2:0)	SEE Marks	60
Credits	03	Exam Hours	03
Module-1			
Laplace Transform: Definition and Laplace transforms of elementary functions (statements only). Laplace transforms of Periodic functions (statement only) and unit-step function – problems. Inverse Laplace Transform: Definition and problems, Convolution theorem to find the inverse Laplace transforms (without Proof) and problems. Solution of linear differential equations using Laplace transforms.			
Module-2			
Fourier Series: Periodic functions, Dirichlet's condition. Fourier series of periodic functions period 2π and arbitrary period. Half range Fourier series. Practical harmonic analysis.			
Module-3			
Fourier Transforms: Infinite Fourier transforms, Fourier sine and cosine transforms. Inverse Fourier transforms. Problems. Difference Equations and Z-Transforms: Difference equations, basic definition, z-transform-definition, Standard z-transforms, Damping and shifting rules, initial value and final value theorems (without proof) and problems, Inverse z-transform and applications to solve difference equations.			
Module-4			
Numerical Solutions of Ordinary Differential Equations(ODE's): Numerical solution of ODE's of first order and first degree- Taylor's series method, Modified Euler's method. Runge -Kutta method of fourth order, Milne's and Adam-Bash forth predictor and corrector method (No derivations of formulae)-Problems.			
Module-5			
Numerical Solution of Second Order ODE's: Runge-Kutta method and Milne's predictor and corrector method. (No derivations of formulae). Calculus of Variations: Variation of function and functional, variational problems, Euler's equation, Geodesics, hanging chain, problems.			
Course outcomes: At the end of the course the student will be able to: <ul style="list-style-type: none">• Solve first and second order ordinary differential equations arising in engineering problems using single step and multistep numerical methods.• Use Laplace transform and inverse Laplace transform in solving differential/ integral equation arising in network analysis, control systems and other fields of engineering.• Demonstrate Fourier series to study the behavior of periodic functions and their applications in system communications, digital signal processing and field theory.• Determine the externals of functional using calculus of variations and solve problems arising in dynamics of rigid bodies and vibrational analysis.• Make use of Fourier transform and Z-transform to illustrate discrete/continuous function arising in wave and heat propagation, signals and systems.			
Question paper pattern:			

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

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Advanced Engineering Mathematics	E. Kreyszig	John Wiley & Sons	10 th Edition, 2016
2	Higher Engineering Mathematics	B. S. Grewal	Khanna Publishers	44 th Edition, 2017
3	Engineering Mathematics	Srimanta Pal et al	Oxford University Press	3 rd Edition, 2016
Reference Books				
1	Advanced Engineering Mathematics	C. Ray Wylie, Louis C. Barrett	McGraw-Hill Book Co	6 th Edition, 1995
2	Introductory Methods of Numerical Analysis	S.S.Sastry	Prentice Hall of India	4 th Edition 2010
3	Higher Engineering Mathematics	B.V. Ramana	McGraw-Hill	11 th Edition,2010
4	A Textbook of Engineering Mathematics	N.P.Bali and Manish Goyal	Laxmi Publications	6 th Edition, 2014
5	Advanced Engineering Mathematics	Chandrika Prasad and Reena Garg	Khanna Publishing,	2018
Web links and Video Lectures:				
1. http://nptel.ac.in/courses.php?disciplineID=111				
2. http://www.class-central.com/subject/math(MOOCs)				
3. http://academicearth.org/				
4. VTU EDUSAT PROGRAMME - 20				

DATA STRUCTURES AND APPLICATIONS			
SEMESTER – III			
Course Code	18CS32	CIE Marks	40
Number of Contact Hours/Week	3:2:0	SEE Marks	60
Total Number of Contact Hours	50	Exam Hours	03
CREDITS –4			
Module 1			Contact Hours
<p>Introduction: Data Structures, Classifications (Primitive & Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, Dynamically allocated arrays.</p> <p>Array Operations: Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials and Sparse Matrices.</p> <p>Strings: Basic Terminology, Storing, Operations and Pattern Matching algorithms. Programming Examples.</p> <p>Textbook 1: Chapter 1: 1.2, Chapter 2: 2.2 - 2.7 Textbook 2: Chapter 1: 1.1 - 1.4, Chapter 3: 3.1 - 3.3, 3.5, 3.7, Ch apter 4: 4.1 - 4.9, 4.14 Reference 3: Chapter 1: 1.4</p> <p>RBT: L1, L2, L3</p>			10
Module 2			
<p>Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression.</p> <p>Recursion - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function.</p> <p>Queues: Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. Multiple Stacks and Queues. Programming Examples.</p> <p>Textbook 1: Chapter 3: 3.1 -3.7 Textbook 2: Chapter 6: 6.1 -6.3, 6.5, 6.7-6.10, 6.12, 6.13</p> <p>RBT: L1, L2, L3</p>			10
Module 3			
<p>Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples</p> <p>Textbook 1: Ch apter 4: 4.1 – 4.6, 4.8, Textbook 2: Ch apter 5: 5.1 – 5.10,</p> <p>RBT: L1, L2, L3</p>			10
Module 4			
<p>Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples</p>			10

Textbook 1: Chapter 5: 5.1 –5.5, 5.7; Textbook 2: Chapter 7: 7.1 – 7.9 RBT: L1, L2, L3	
Module 5	
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, Address Calculation Sort. Hashing: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. Files and Their Organization: Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing Textbook 1: Chapter 6 : 6.1 –6.2, Chapter 7:7.2, Chapter 8 : 8.1-8.3 Textbook 2: Chapter 8 : 8.1 – 8.7, Chapter 9 : 9.1-9.3, 9.7, 9.9 Reference 2: Chapter 16 : 16.1 - 16.7 RBT: L1, L2, L3	10
Course Outcomes: The student will be able to :	
<ul style="list-style-type: none"> • Apply the fundamental concepts of data structures and their applications essential for programming/problem solving. • Make use of stacks to evaluate mathematical expressions and queues for mazing problem. • Choose linked lists to implement of lists, stacks, queues, polynomials and sparse matrix. • Construct various types of trees using linked lists and apply tree traversal methods for expressions evaluation. • Utilize BFS, DFS, searching, sorting, hashing and files concepts to develop various applications. 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • 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. 	
Textbooks:	
<ol style="list-style-type: none"> 1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014. 2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014. 	
Reference Books:	
<ol style="list-style-type: none"> 1. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014. 2. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012. 3. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Ed, McGraw Hill, 2013 4. A M Tenenbaum, Data Structures using C, PHI, 1989 5. Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996. 	

ANALOG AND DIGITAL ELECTRONICS SEMESTER – III			
Course Code	18CS33	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Module 1			Contact Hours
Photodiodes, Light Emitting Diodes and Optocouplers ,BJT Biasing :Fixed bias ,Collector to base Bias , voltage divider bias, Operational Amplifier Application Circuits: Multivibrators using IC-555, Peak Detector, Schmitt trigger, Active Filters, Non-Linear Amplifier, Relaxation Oscillator, Current-to-Voltage and Voltage-to-Current Converter , Regulated Power Supply Parameters, adjustable voltage regulator ,D to A and A to D converter. Text Book 1 :Part A:Chapter 2(Section 2.9,2.10,2.11), Chapter 4(Section 4.2 ,4.3,4.4),Chapter 7 (section (7.2,7.3.1,7.4,7.6 to 7.11), Chapter 8 (section (8.1,8.5), Chapter 9 RBT: L1, L2			08
Module 2			
Karnaugh maps: minimum forms of switching functions, two and three variable Karnaugh maps, four variable karnaugh maps, determination of minimum expressions using essential prime implicants, Quine-McClusky Method: determination of prime implicants, The prime implicant chart, petricks method, simplification of incompletely specified functions, simplification using map-entered variables Text book 1:Part B: Chapter 5 (Sections 5.1 to 5.4) Chapter 6(Sections 6.1 to 6.5) RBT: L1, L2			08
Module 3			
Combinational circuit design and simulation using gates: Review of Combinational circuit design, design of circuits with limited Gate Fan-in ,Gate delays and Timing diagrams, Hazards in combinational Logic, simulation and testing of logic circuits Multiplexers, Decoders and Programmable Logic Devices: Multiplexers, three state buffers, decoders and encoders, Programmable Logic devices, Programmable Logic Arrays, Programmable Array Logic. Text book 1:Part B: Chapter 8,Chapter 9 (Sections 9.1 to 9.6) RBT: L1, L2			08
Module 4			
Introduction to VHDL: VHDL description of combinational circuits, VHDL Models for multiplexers, VHDL Modules. Latches and Flip-Flops: Set Reset Latch, Gated Latches, Edge-Triggered D Flip Flop 3,SR Flip Flop, J K Flip Flop, T Flip Flop, Flip Flop with additional inputs, Asynchronous Sequential Circuits Text book 1:Part B: Chapter 10(Sections 10.1 to 10.3),Chapter 11 (Sections 11.1 to 11.9) RBT: L1, L2			08

Module 5	
Registers and Counters: Registers and Register Transfers, Parallel Adder with accumulator, shift registers, design of Binary counters, counters for other sequences, counter design using SR and J K Flip Flops, sequential parity checker, state tables and graphs Text book 1:Part B: Chapter 12(Sections 12.1 to 12.5),Chapter 13(Sections 13.1,13.3) RBT: L1, L2	08
Course Outcomes: The student will be able to :	
<ul style="list-style-type: none"> • Design the applications of analog circuits using photo devices, timer IC, power supply and regulator and IC op-amp for various applications such as Schmitt trigger, relaxation oscillator etc.. • Choose the Combinational Logic circuits and simplification techniques such as Karnaugh Maps, Quine McClusky Techniques for designing various digital circuits. • Construct different circuits using Decoders, Encoders, Multiplexers, Adders and Subtractors. • Make use of the latches, Flip-Flops, HDL programs for constructing and simulating sequential circuits. • Obtain the steps to design counters and registers 	
Question Paper Pattern:	
<ul style="list-style-type: none"> • 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. 	
Textbooks:	
1. Charles H Roth and Larry L Kinney, Analog and Digital Electronics, Cengage Learning,2019	
Reference Books:	
<ol style="list-style-type: none"> 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. 3. M. Morris Mani, Digital Design, 4th Edition, Pearson Prentice Hall, 2008. 4. David A. Bell, Electronic Devices and Circuits, 5th Edition, Oxford University Press, 2008 	

COMPUTER ORGANIZATION SEMESTER – III			
Course Code	18CS34	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Module 1			Contact Hours
Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions Text book 1: Chapter1 – 1.3, 1.4, 1.6 (1.6.1-1.6.4, 1.6.7), Chapter2 – 2.2 to 2.10 RBT: L1, L2, L3			08
Module 2			
Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB. Text book 1: Chapter4 – 4.1, 4.2, 4.4, 4.5, 4.6, 4.7 RBT: L1, L2, L3			08
Module 3			
Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations. Text book 1: Chapter5 – 5.1 to 5.4, 5.5 (5.5.1, 5.5.2), 5.6 RBT: L1, L2, L3			08
Module 4			
Arithmetic: Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division. Text book 1: Chapter2-2.1, Chapter6 – 6.1 to 6.6 RBT: L1, L2, L3			08
Module 5			
Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Micro programmed Control. Pipelining: Basic concepts of pipelining, Text book 1: Chapter7, Chapter8 – 8.1 RBT: L1, L2, L3			08
Course Outcomes: The student will be able to : <ul style="list-style-type: none"> • Able to outline the software engineering principles and illustrate the activities involved in building large software and also illustrating the process of requirements, requirements classification. • Demonstrate Object Orientation Modelling Concepts and Class Modelling • Analyze the system models, examine the object oriented design patterns and list out the open source development tools • To choose the appropriate software testing type, also identify the significance of software maintenance. • To choose the right software pricing and measurements of software metrics. Also to identify the software quality parameters and standards 			

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.

Textbooks:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, 5th Edition, Tata McGraw Hill, 2002. (Listed topics only from Chapters 1, 2, 4, 5, 6, 7, 8, 9 and 12)

Reference Books:

1. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015.

SOFTWARE ENGINEERING SEMESTER – III			
Course Code	18CS35	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Module 1			Contact Hours
<p>Introduction: Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics. Case Studies.</p> <p>Software Processes: Models: Waterfall Model (Sec 2.1.1), Incremental Model (Sec 2.1.2) and Spiral Model (Sec 2.1.3). Process activities.</p> <p>Requirements Engineering: Requirements Engineering Processes (Chap 4). Requirements Elicitation and Analysis (Sec 4.5). Functional and non-functional requirements (Sec 4.1). The software Requirements Document (Sec 4.2). Requirements Specification (Sec 4.3). Requirements validation (Sec 4.6). Requirements Management (Sec 4.7).</p> <p>RBT: L1, L2, L3</p>			08
Module 2			
<p>What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. Introduction, Modelling Concepts and Class Modelling: What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. Class Modelling: Object and Class Concept; Link and associations concepts; Generalization and Inheritance; A sample class model; Navigation of class models;</p> <p>Textbook 2: Ch 1,2,3.</p> <p>RBT: L1, L2 L3</p>			08
Module 3			
<p>System Models: Context models (Sec 5.1). Interaction models (Sec 5.2). Structural models (Sec 5.3). Behavioral models (Sec 5.4). Model-driven engineering (Sec 5.5).</p> <p>Design and Implementation: Introduction to RUP (Sec 2.4), Design Principles (Chap 7). Object-oriented design using the UML (Sec 7.1). Design patterns (Sec 7.2). Implementation issues (Sec 7.3). Open source development (Sec 7.4).</p> <p>RBT: L1, L2, L3</p>			08
Module 4			

<p>Software Testing: Development testing (Sec 8.1), Test-driven development (Sec 8.2), Release testing (Sec 8.3), User testing (Sec 8.4). Test Automation (Page no 212).</p> <p>Software Evolution: Evolution processes (Sec 9.1). Program evolution dynamics (Sec 9.2). Software maintenance (Sec 9.3). Legacy system management (Sec 9.4).</p> <p>RBT: L1, L2, L3</p>	08
Module 5	
<p>Project Planning: Software pricing (Sec 23.1). Plan-driven development (Sec 23.2). Project scheduling (Sec 23.3): Estimation techniques (Sec 23.5). Quality management: Software quality (Sec 24.1). Reviews and inspections (Sec 24.3). Software measurement and metrics (Sec 24.4). Software standards (Sec 24.2)</p> <p>RBT: L1, L2, L3</p>	08
<p>Course Outcomes: The student will be able to :</p>	
<ul style="list-style-type: none"> • Able to outline the software engineering principles and illustrate the activities involved in building large software and also illustrating the process of requirements, requirements classification. • Demonstrate Object Orientation Modelling Concepts and Class Modelling • Analyze the system models, examine the object oriented design patterns and list out the open source development tools • To choose the appropriate software testing type, also identify the significance of software maintenance. • To choose the right software pricing and measurements of software metrics. Also to identify the software quality parameters and standards 	
<p>Question Paper Pattern:</p>	
<ul style="list-style-type: none"> • 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. 	
<p>Textbooks:</p>	
<ol style="list-style-type: none"> 1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24) 2. Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML,2nd Edition, Pearson Education,2005. 	
<p>Reference Books:</p>	
<ol style="list-style-type: none"> 1. Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill. 2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India 	

DISCRETE MATHEMATICAL STRUCTURES SEMESTER – III			
Course Code	18CS36	CIE Marks	40
Number of Contact Hours/Week	3:0:0	SEE Marks	60
Total Number of Contact Hours	40	Exam Hours	03
CREDITS –3			
Module 1			Contact Hours
Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. Fundamentals of Logic contd.: The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems. Text book 1: Chapter2 RBT: L1, L2, L3			08
Module 2			
Properties of the Integers: The Well Ordering Principle – Mathematical Induction, Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition. Text book 1: Chapter4 – 4.1, Chapter1 RBT: L1, L2, L3			08
Module 3			
Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions. Relations: Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. Text book 1: Chapter5 , Chapter7 – 7.1 to 7.4 RBT: L1, L2, L3			08
Module 4			
The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials. Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients. Text book 1: Chapter8 – 8.1 to 8.4, Chapter10 – 10.1, 10.2 RBT: L1, L2, L3			08
Module 5			
Introduction to Graph Theory: Definitions and Examples, Sub graphs, Complements, and Graph Isomorphism, Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes Text book 1: Chapter11 – 11.1 to 11.2 Chapter12 – 12.1 to 12.4 RBT: L1, L2, L3			08
Course Outcomes: The student will be able to: <ul style="list-style-type: none"> • Interpret propositional and predicate logic in knowledge representation and truth verification. • Demonstrate the properties of integers and fundamental principle of counting in discrete structures. • Utilize the understandings of relations and functions and be able to determine their properties • Solve the problems using the concept of graph theory and trees properties • Solve problems using recurrence relations and Principle of Inclusion and Exclusion 			

Question Paper Pattern:
<ul style="list-style-type: none">• 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.
Textbooks:
1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education. 2004.
Reference Books:
<ol style="list-style-type: none">1. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics – A Concept based approach, Universities Press, 20162. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.3. Jayant Ganguly: A Treatise on Discrete Mathematical Structures, Sanguine-Pearson, 2010.4. D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004.5. Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008.

**ANALOG AND DIGITAL ELECTRONICS LABORATORY
SEMESTER – III**

Course Code	18CSL37	CIE Marks	40
Number of Contact Hours/Week	0:2:2	SEE Marks	60
Total Number of Lab Contact Hours	36	Exam Hours	03

Credits – 2

Descriptions (if any):

- Simulation packages preferred: Multisim, Modelsim, PSpice or any other relevant.
- For Part A (Analog Electronic Circuits) students must trace the wave form on Tracing sheet / Graph sheet and label trace.
- Continuous evaluation by the faculty must be carried by including performance of a student in both hardware implementation and simulation (if any) for the given circuit.
- A batch not exceeding 4 must be formed for conducting the experiment. For simulation individual student must execute the program.

Laboratory Programs:

PART A (Analog Electronic Circuits)

1.	Design an astable multivibrator circuit for three cases of duty cycle (50%, <50% and >50%) using NE 555 timer IC. Simulate the same for any one duty cycle.
2.	Using ua 741 Opamp, design a 1 kHz Relaxation Oscillator with 50% duty cycle. And simulate the same.
3.	Using ua 741 opamp, design a window comparator for any given UTP and LTP. And simulate the same.

PART B (Digital Electronic Circuits)

4.	Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates. And implement the same in HDL.
5.	Given a 4-variable logic expression, simplify it using appropriate technique and realize the simplified logic expression using 8:1 multiplexer IC. And implement the same in HDL.
6.	Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table. And implement the same in HDL.
7.	Design and implement code converter I) Binary to Gray (II) Gray to Binary Code using basic gates.
8.	Design and implement a mod-n ($n < 8$) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
9.	Design and implement an asynchronous counter using decade counter IC to count up from 0 to n ($n \leq 9$) and demonstrate on 7-segment display (using IC-7447)

Laboratory Outcomes: The student should be able to:

- Utilize Operational Amplifier and timers for different applications also make use of simulation package to design circuits
- Build window comparator and simulate.
- Choose the Combinational Logic circuits for realizing adders, subtractors and multiplexers and also simulate the same
- Design MSJK Flip Flop, also make use of simulation package to design circuits.
- Construct code converters circuits, synchronous and asynchronous counters.

Conduct of Practical Examination:

- Experiment distribution
 - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
 - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (*Courseed to change in accordance with university regulations*)
 - a) For laboratories having only one part – Procedure + Execution + Viva-Voce: $15+70+15 = 100$ Marks
 - b) For laboratories having PART A and PART B
 - i. Part A – Procedure + Execution + Viva = $6 + 28 + 6 = 40$ Marks
 - ii. Part B – Procedure + Execution + Viva = $9 + 42 + 9 = 60$ Marks

DATA STRUCTURES LABORATORY
(Effective from the academic year 2018 -2019)
SEMESTER – III

Course Code	18CSL38	CIE Marks	40
Number of Contact Hours/Week	0:2:2	SEE Marks	60
Total Number of Lab Contact Hours	36	Exam Hours	03
Credits – 2			
Descriptions (if any):			
<ul style="list-style-type: none"> • Implement all the programs in ‘C / C++’ Programming Language and Linux / Windows as OS. 			
Programs List:			
1.	Design, Develop and Implement a menu driven Program in C for the following array operations. <ol style="list-style-type: none"> a. Creating an array of N Integer Elements b. Display of array Elements with Suitable Headings c. Inserting an Element (ELEM) at a given valid Position (POS) d. Deleting an Element at a given valid Position (POS) e. Exit. Support the program with functions for each of the above operations.		
2.	Design, Develop and Implement a Program in C for the following operations on Strings. <ol style="list-style-type: none"> a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR Support the program with functions for each of the above operations. Don't use Built-in functions.		
3.	Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) <ol style="list-style-type: none"> a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack e. Display the status of Stack f. Exit Support the program with appropriate functions for each of the above operations		
4.	Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.		
5.	Design, Develop and Implement a Program in C for the following Stack Applications <ol style="list-style-type: none"> a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ b. Solving Tower of Hanoi problem with n disks 		

6.	<p>Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)</p> <ol style="list-style-type: none"> Insert an Element on to Circular QUEUE Delete an Element from Circular QUEUE Demonstrate Overflow and Underflow situations on Circular QUEUE Display the status of Circular QUEUE Exit <p>Support the program with appropriate functions for each of the above operations</p>
7.	<p>Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: <i>USN, Name, Programme, Sem, PhNo</i></p> <ol style="list-style-type: none"> Create a SLL of N Students Data by using <i>front insertion</i>. Display the status of SLL and count the number of nodes in it Perform Insertion / Deletion at End of SLL Perform Insertion / Deletion at Front of SLL(Demonstration of stack) Exit
8.	<p>Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: <i>SSN, Name, Dept, Designation, Sal, PhNo</i></p> <ol style="list-style-type: none"> Create a DLL of N Employees Data by using <i>end insertion</i>. Display the status of DLL and count the number of nodes in it Perform Insertion and Deletion at End of DLL Perform Insertion and Deletion at Front of DLL Demonstrate how this DLL can be used as Double Ended Queue. Exit
9.	<p>Design, Develop and Implement a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes</p> <ol style="list-style-type: none"> Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2y^2z - 4yz^5 + 3x^3yz + 2xy^5z - 2xyz^3$ Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) <p>Support the program with appropriate functions for each of the above operations</p>
10.	<p>Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers .</p> <ol style="list-style-type: none"> Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 Traverse the BST in Inorder, Preorder and Post Order Search the BST for a given element (KEY) and report the appropriate message Exit
11.	<p>Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities</p> <ol style="list-style-type: none"> Create a Graph of N cities using Adjacency Matrix. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method
12.	<p>Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function $H: K \rightarrow L$ as $H(K) = K \text{ mod } m$ (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.</p>
<p>Laboratory Outcomes: The student should be able to:</p>	

- Experiment with array operations and string application programs.
- Construct the programs to implement stacks, queues and their applications.
- Develop the programs to implement various operations of linked lists and their applications.
- Make use of tree concepts to implement programs for their applications
- Apply DFS/BFS method for graph traversals and linear probing approach for hashing programs.

Conduct of Practical Examination:

- Experiment distribution
 - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
 - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (*Courseed to change in accordance with university regulations*)
 - c) For laboratories having only one part – Procedure + Execution + Viva-Voce: $15+70+15 = 100$ Marks
 - d) For laboratories having PART A and PART B
 - i. Part A – Procedure + Execution + Viva = $6 + 28 + 6 = 40$ Marks
 - ii. Part B – Procedure + Execution + Viva = $9 + 42 + 9 = 60$ Marks

