ENGINEERING MATHEMATICS-III

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15MAT31**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Understand and use of analytical and numerical methods in different engineering fields
- Understand and apply Fourier Series
- Understand and use of Fourier transforms and Z-Transforms
- Use of statistical methods in curve fitting applications
- Use of numerical methods to solve algebraic and transcendental equations, vector integration and calculus of variation

Module -1	**Teaching Hours**	**RBT Levels**
Fourier Series: Periodic functions, Dirichlet's condition, Fourier Series of Periodic functions with period 2π and with arbitrary period 2c, Fourier series of even and odd functions, Half range Fourier Series, practical Harmonic analysis. Complex Fourier series	**10Hours**	**L1, L2, L3,L4**
Module -2		
Fourier Transforms: Infinite Fourier transforms, Fourier Sine and Cosine transforms, Inverse transform. **Z-transform:** Difference equations, basic definition, z-transform-definition, Standard z-transforms, Damping rule, Shifting rule, Initial value and final value theorems (without proof) and problems, Inverse z-transform. Applications of z-transforms to solve difference equations..	**10 Hours**	**L1, L2, L3,L4**
Module - 3		
Statistical Methods: Correlation and rank Correlation coefficients, Regression and Regression coefficients, lines of regression - problems **Curve fitting:** Curve fitting by the method of least squares, Fitting of the curves of the form, $y = ax + b, y = ax^2 + bx + c, y = ae^{bx}, y = ax^b$.**Numerical Methods:** Numerical solution of algebraic and transcendental equations by: Regular-falsi method, Secant method, Newton - Raphson method and Graphical method. .	**10 Hours**	**L1, L2, L3,L4**
Module-4		
Finite differences: Forward and backward differences, Newton's forward and backward interpolation formulae. Divided differences-Newton's divided difference formula. Lagrange's interpolation formula and inverse interpolation formula. Central Difference-Stirling's and Bessel's formulae (all formulae without proof)-Problems. **Numerical integration**: Simpson's 1/3, 3/8 rule, Weddle's rule (without proof) -Problems	**10 Hours**	**L1, L2, L3,L4**
Module-5		

Vector integration: Line integrals-definition and problems, surface and volume integrals-definition, Green's theorem in a plane, Stokes and Gauss-divergence theorem(without proof) and problems. **Calculus of Variations:** Variation of function and Functional, variational problems, Euler's equation, Geodesics, minimal surface of revolution, hanging chain, problems	**10 Hours**	**L1, L2, L3,L4**

Course outcomes:

After Studying this course, students will be able to

- know the use of periodic signals and Fourier series to analyze circuits
- explain the general linear system theory for continuous-time signals and systems using the Fourier Transform
- Analyse discrete-time systems using convolution and the z-transform
- use appropriate numerical methods to solve algebraic and transcendental equations and also to calculate a definite integral
- Use curl and divergence of a vector function in three dimensions, as well as apply the Green's Theorem, Divergence Theorem and Stokes' theorem in various applications
- Solve the simple problem of the calculus of variations

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Life-Long Learning
4. Conduct Investigations of Complex Problems

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. B. S. Grewal," Higher Engineering Mathematics", Khanna publishers, 42nd edition, 2013.
2. B.V.Ramana "Higher Engineering M athematics" Tata McGraw-Hill, 2006

Reference Books:

1. N P Bali and Manish Goyal, "A text book of Engineering mathematics" , Laxmi publications, latest edition.
2. Kreyszig, "Advanced Engineering Mathematics " - 9th edition, Wiley,
3. H. K Dass and Er. RajnishVerma ,"Higher Engineerig Mathematics", S. Chand, 1st ed,

Analog and Digital Electronics

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS32**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Recall and Recognize construction and characteristics of JFETs and MOSFETs.
- Describe, Differentiate and Apply JFETs and MOSFETs
- Define, Demonstrate and Analyse Operational Amplifier circuits and their applications
- Describe, Illustrate and Analyse Combinational Logic circuits, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques.
- Define, Describe and Design Decoders, Encoders, Digital multiplexers, Adders and Subtractors, Binary comparators, Latches and Master-Slave Flip-Flops.
- Describe, Demonstrate, Analyse and Design Synchronous and Asynchronous Sequential Circuits, State diagrams, Registers and Counters, A/D and D/A converters.

Module -1	**Teaching Hours**	**RBT Levels**
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, **Operational Amplifier Application Circuits**:Peak Detector Circuit, Comparator, Active Filters, Non-Linear Amplifier, Relaxation Oscillator, Current-To-Voltage Converter, Voltage-To-Current Converter. **(Text book 1:- Ch5:5.2, 5.3, 5.5, 5.8,5.9, 5.1.Ch13: 13.10.Ch 16: 16.3, 16.4.**	**10 Hours**	**L1,L2, L3**
Module -2		
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 Quine-McCluskyMethod, Hazards and Hazard covers, HDL Implementation Models. **Text book 2:- Ch2: 2.4,2.5. Ch3: 3.2 to 3.11.**	**10 Hours**	**L1,L2, L3**
Module - 3		

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 FLIP-FLOPs. **Text book 2:- Ch 4:- 4.1 to 4.9, 4.11, 4.12, 4.14.Ch6:-6.7, 6.10.Ch8:- 8.1 to 8.5.**	**10 Hours**	**L2, L3, L4**
Module-4		
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, Synchronous Counters, Changing the Counter Modulus. (**Text book 2:- Ch 8: 8.6, 8.8, 8.9, 8.10, 8.13. Ch 9: 9.1 to 9.8. Ch 10: 10.1 to**	**10 Hours**	**L2, L3, L4, L6**
Module-5		
Counters: Decade Counters, Pre settable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL. **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, A/D Accuracy and Resolution. **Text book 2:- Ch 10: 10.5 to 10.9. Ch 12: 12.1 to 12.10**	**10 Hours**	**L2, L3, L4, L6**
Course outcomes:		
After studying this course, students will be able to: • 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. - Working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters • Analyse the performance of - JFETs and MOSFETs , Operational Amplifier circuits - Simplification Techniques using Karnaugh Maps, Quine McClusky Technique. - Synchronous and Asynchronous Sequential Circuits. • Apply the knowledge gained in the design of Counters, Registers and A/D & D/A converters		
Graduate Attributes (as per NBA) 1. Engineering Knowledge 2. Design/Development of Solutions(partly) 3. Modern Tool Usage		

4. Problem Analysis

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, VarshaAgarwal: Electronic Devices and Circuits, Wiley, 2012.
2. Donald P Leach, Albert Paul Malvino&GoutamSaha: Digital Principles and Applications, 7th Edition, Tata McGraw Hill, 2014

Reference Books:

1. Stephen Brown, ZvonkoVranesic: 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, 10th Edition, Pearson, 2008.

DATA STRUCTURES AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS33**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Understand, Practice and Assimilate fundamentals of data structures and their applications essential for programming/problem solving
- Describe, Analyze, Design and Evaluate the Linear Data Structures: Stack, Queues, Lists
- Describe, Analyze, Design and Evaluate the Non-Linear Data Structures: Trees, Graphs
- Describe, Analyze, Design and Evaluate the sorting & searching algorithms
- Assess appropriate data structure during program development/Problem Solving

Module -1	**Teaching Hours**	**RBT Levels**
Introduction to Data Structures, Classification of Data Structures: Primitive and Non-Primitive, Linear and Nonlinear; Data structure Operations: Create, Insert, Delete, Search, Sort, Merge, Traversal. Review of Structures, Unions and Pointers, Self Referential Structures. ***Arrays:*** Definition, Representation, Operations - Insert, Delete, Simple Merge, Search, Sort; Multidimensional Arrays; Applications of Arrays. ***Strings:*** Definition, Representation, Operations, and String manipulation Applications. Dynamic Memory Management Functions - *malloc, calloc, realloc, free.*, Programming Examples.	**10Hours**	**L1, L2**
Module -2		
Linear Data Structures and their Sequential Storage Representation: ***Stack:*** Definition, Representation, Operations and Applications: Polish and reverse polish expressions, Infix to postfix conversion, evaluation of postfix expression, infix to prefix, postfix to infix conversion; Recursion - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Binomial Co-efficient(***nCr***), Ackerman's Recursive function. ***Queue:*** Definition, Representation, Operations, Queue Variants: Circular Queue, Priority Queue, Double Ended Queue; Applications of Queues. Programming Examples.	**10 Hours**	**L1, L2, L3, L4, L6**
Module - 3		
Linear Data Structures and their Linked Storage Representation: ***Linked List:*** Definition, Representation, Operations, Types: Singly Linked List, Doubly Linked list, Circular linked list. Linked implementation of Stack, Queue and its variants - Double Ended, Priority queues. Applications of Linked lists - Polynomial Manipulation, multiprecision arithmetic, Symbol table organizations, Sparse matrix representation with multilinked data structure. Programming Examples - length of a list, Merging two lists, removing duplicates, reversing a list, union and intersection of two lists etc.,	**10 Hours**	**L2, L3, L4, L6**

Module-4		
Nonlinear Data Structures: ***Trees***: Definitions, Terminologies, Array and linked Representation of Binary Trees, Types- Complete/full, Almost Complete, Strictly, Skewed; Traversal methods - Inorder, postorder, preorder; Binary Search Trees - Creation, Insertion, Deletion, Traversal, Searching; Expression tree, Threaded binary tree, Conversion of General Trees to Binary Trees, Constructing BST from traversal orders; Applications Of Trees: Evaluation of Expression, Tree based Sorting. Programming Examples	**10 Hours**	**L2, L3, L4, L6**
Module-5		
Graph: 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*: The Hash Table organizations, Hashing Functions, Static and Dynamic Hashing, Collision-Resolution Techniques, Programming Examples. ***File Structures:*** Definitions and Concepts, Types, File Organizations - Sequential, Indexed Sequential, Random Access.	**10 Hours**	**L2, L3, L4, L6**

Course outcomes:

After studying this course, students will be able to:

- Acquire knowledge of
  - Various types of data structures, operations and algorithms
  - Sorting and searching operations
  - File structures
- Analyse the performance of
  - Stack, Queue, Lists, Trees, Graphs, Searching and Sorting techniques
- Implement all the applications of Data structures in a high-level language
- Design and apply appropriate data structures for solving computing problems.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Design/Development of Solutions
3. Conduct Investigations of Complex Problems
4. Problem Analysis

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 SartajSahni, 2nd edition, 2014, Universities Press
2. Data Structures: A Pseudo-code approach with C - Gilberg&Forouzan, 2nd edition, 2014, Cengage Learning

Reference Books:

1. Data Structures using C, second edition, Reemathareja, Oxford press
2. Data Structures - Seymour Lipschutz, Schaum's Outlines, revised 1st edition,McGraw Hill
3. An Introduction to Data Structures with Applications- Jean-Paul Tremblay & Paul G. Sorenson, 2nd Edition, 2013, McGraw Hill
4. Data Structures using C - A M Tenenbaum, Pearson
5. Data Structures and Program Design in C - Robert Kruse, PHI

COMPUTER ORGANIZATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	15CS34	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS - 04

Course objectives: This course will enable students to

- Understand the basics of computer organization: structure and operation of computers and their peripherals.
- Understand the concepts of programs as sequences or machine instructions.
- Expose different ways of communicating with I/O devices and standard I/O interfaces.
- Describe hierarchical memory systems including cache memories and virtual memory.
- Describe arithmetic and logical operations with integer and floating-point operands.
- Understand basic processing unit and organization of simple processor, concept of pipelining and other large computing systems.

Module -1	Teaching Hours	RBT Levels
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	10Hours	L1, L2
Module -2		
Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB.	10 Hours	L1, L2
Module - 3		
Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories, Secondary Storage.	10 Hours	L1, L2, L3
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, Floating-point Numbers and Operations.	10 Hours	L1, L2, L3, L4
Module-5		

Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Micro programmed Control. Pipelining, Embedded Systems and Large Computer Systems: Basic Concepts of pipelining, Examples of Embedded Systems, Processor chips for embedded applications, Simple Microcontroller, Forms of parallel processing, Array Processors, The structure of General-Purpose Multiprocessors.	10 Hours	L1, L2, L4, L6

Course outcomes:

After studying this course, students will be able to:

- Acquire knowledge of
  - The basic structure of computers & machine instructions and programs, Addressing Modes, Assembly Language, Stacks, Queues and Subroutines.
  - Input/output Organization such as accessing I/O Devices, Interrupts.
  - Memory system basic Concepts, Semiconductor RAM Memories, Static memories, Asynchronous DRAMS, Read Only Memories, Cache Memories and Virtual Memories.
  - Some Fundamental Concepts of Basic Processing Unit, Execution of a Complete Instruction, Multiple Bus Organization, Hardwired Control and Micro programmed Control.
  - Pipelining, embedded and large computing system architecture.
- Analyse and design arithmetic and logical units.
- Apply the knowledge gained in the design of Computer.
- Design and evaluate performance of memory systems
- Understand the importance of life-long learning

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Life-Long Learning

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, ZvonkoVranesic, SafwatZaky: Computer Organization, 5th Edition, Tata McGraw Hill, 2002. (Listed topics only from Chapters 1, 2, 4, 5, 6, 7, 8, 9 and12)

Reference Books:

1. William Stallings: Computer Organization & Architecture, 7th Edition, PHI, 2006.

DISCRETE MATHEMATICAL STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS35**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Prepare for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science.
- Understand and apply logic, relations, functions, basic set theory, countability and counting arguments, proof techniques,
- Understand and apply mathematical induction, combinatorics, discrete probability, recursion, sequence and recurrence, elementary number theory
- Understand and apply graph theory and mathematical proof techniques.

Module -1	**Teaching Hours**	**RBT Levels**
Set Theory: Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams, A First Word on Probability, Countable and Uncountable Sets. **Fundamentals of Logic:** Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference.	**10Hours**	**L2, L3**
Module -2		
Fundamentals of Logic *contd.*: The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems, **Properties of the Integers:** Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions	**10 Hours**	**L3, L4**
Module - 3		
Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions – Stirling Numbers of the Second Kind, Special Functions, The Pigeon-hole Principle, Function Composition and Inverse Functions.	**10 Hours**	**L3,L4, L5**
Module-4		
Relations *contd.*: Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions	**10 Hours**	**L3,L4, L5**
Module-5		

Groups: Definitions, properties, Homomrphisms, Isomorphisms, Cyclic Groups, Cosets, and Lagrange's Theorem. **Coding Theory and Rings:** Elements of Coding Theory, The Hamming Metric, The Parity Check, and Generator Matrices. **Group Codes:** Decoding with Coset Leaders, Hamming Matrices. **Rings and Modular Arithmetic:** The Ring Structure – Definition and Examples, Ring Properties and Substructures, The Integer modulo n.	**10 Hours**	**L3,L4, L5**

Course outcomes:

After studying this course, students will be able to:

1. Verify the correctness of an argument using propositional and predicate logic and truth tables.
2. Demonstrate the ability to solve problems using counting techniques and combinatorics in the context of discrete probability.
3. Solve problems involving recurrence relations and generating functions.
4. Perform operations on discrete structures such as sets, functions, relations, and sequences.
5. Construct proofs using direct proof, proof by contraposition, proof by contradiction, proof by cases, and mathematical induction.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Conduct Investigations of Complex Problems
4. Design/Development of Solutions

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.Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, , 5th Edition, Pearson Education. 2004.

(Chapter 3.1, 3.2, 3.3, 3.4, Appendix 3, Chapter 2, Chapter 4.1, 4.2, Chapter 5.1 to 5.6, Chapter 7.1 to 7.4, Chapter 16.1, 16.2, 16.3, 16.5 to 16.9, and Chapter 14.1, 14.2, 14.3).

Reference Books:

1. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.
2. JayantGanguly: A Treatise on Discrete Mathematical Structures, Sanguine-Pearson, 2010.
3. D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004.
4. Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008.

UNIX AND SHELL PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS361**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Understand the UNIX Architecture, File systems and use of basic Commands.
- Use of editors and Networking commands.
- Understand Shell Programming and to write shell scripts.
- Understand and analyze UNIX System calls, Process Creation, Control & Relationship.

Module -1	**Teaching Hours**	**RBT Levels**
Introduction - Why UNIX? , Computer System, The UNIX Environment, UNIX Structure, Accessing Unix, Commands, Common Commands, Other Useful Commands. **File Systems-** Filenames, File types, Regular Files, Directories, File System Implementation, Operations Unique to Directories, Operations Unique to Regular Files, Operations Common to Both. **Security and File Permission** – Users and Groups, Security Levels, Changing permissions, User masks , Changing Ownership and group.	**08Hours**	**L1, L2**
Module -2		
The Basic vi Editor–Editor Concepts , The Vi editor , Modes, Commands, Command Categories, Local Commands in vi, Range commands in vi, Global Commands in vi, Rearrange Text in vi, ex editor. **Introduction to Shells**- Unix Session , Standard Streams , Redirection, Pipes , tee command , Command execution , Quotes , Command substitution, Job Control, Aliases, Variables, predefined variables, Options, Shell/Environment Customization.	**08Hours**	**L1, L2, L5, L6**
Module - 3		
Communications – User Communication, Electronic Mail, Remote Access, File Transfer. **Interactive Korn Shell** – Korn Shell Features, Two Special Files , Variables, Output, Input, exit status of a command, eval command , Environmental variables, options, Startup Scripts , Command History, Command execution process. **Korn Shell Programming** – Basic Script Concepts, Expressions, Decisions: Making Selections, Repetition, Special Parameters and variables, Changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.	**08Hours**	**L1, L2, L5, L6**
Module-4		

File I/O- Introduction**,** File Descriptors, open Function, creat Function, close Function , seek Function, read Function, write Function, I/O Efficiency , File Sharing , Atomic Operations, dup and dup2 Functions, sync, fsync and fdatasync Functions ,fctnl Functions, ioctl Functions , /dev/fd. **UNIX Processes:** The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables.	**08Hours**	**L1, L2, L5, L6**
Module-5		
Process Control : Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times .**Process Relationships**: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp, tcsetpgrp and tcgetsid Functions, Job Control.	**08Hours**	**L1, L2, L5, L6**

Course outcomes:

After studying this course, students will be able to:

- Explain multi user OS UNIX and its basic features
- Interpret UNIX Commands, Shell basics, and shell environments
- Design and develop shell programming, communication, System calls and terminology.
- Design and develop UNIX File I/O and UNIX Processes.
- Understand UNIX process control, relationships, commands and utilities

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Environment and Sustainability
3. Design/Development of Solutions

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. Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. (Chapters- 1,2, 3, 4, 5, 7,8, 13, 14) 2009.
2. W. Richard Stevens, Stephen A Rago: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education.(Chapters 3,7.1 to 7.9, 8, 9.1 to 9.8) .2009

Reference Books:

1. Sumitabha Das: UNIX – Concepts and Applications,4th Edition, Tata McGraw Hill.
2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2nd Edition , Wiley,2014
3. M.G. Venkateshmurthy: UNIX & Shell Programming**,** Pearson Education.

PROBABILITY AND STATISTICS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS362**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Acquire knowledge of Probability theory and Statistical methods and their applications
- Develop analytical capability
- Applying Engineering and Technology
- Solve the real world problems.

Module -1	**Teaching Hours**	**RBT Levels**
Statistics and Probability: Overview: Statistical Inference, Samples, Populations, and the Role of Probability, Sampling Procedures; Collection of Data, Discrete and Continuous Data, Probability: Sample Space and Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Multiplicative Rule, Bayes' Rule.	**08Hours**	**L2,L3, L4**
Module -2		
Random Variables, Distributions and Expectations: Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions, Joint Probability Distributions, Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's theorem.	**08 Hours**	**L2,L3, L4**
Module - 3		
Probability Distributions: Binomial and Multinomial Distributions, Hypergeometric Distribution, Negative Binomial and Geometric Distributions, Poisson Distribution and the Poisson Process, Continuous Uniform Distribution, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Gamma and Exponential Distributions, Chi-Squared Distribution.	**08 Hours**	**L2,L3, L4**
Module-4		
Sampling Distributions: Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem, Sampling Distribution of S2, t-Distribution, F-Distribution.	**08 Hours**	**L2,L3, L4**
Module-5		

Estimation and Hypothesis: Statistical Inference, Classical Methods of Estimation, Single Sample: Estimating the Mean, Statistical Hypotheses: General Concepts, Testing a Statistical Hypothesis, One- and Two-Tailed Tests, The Use of P-Values for Decision Making in Testing Hypotheses.	**08 Hours**	**L2,L3, L4**

Course outcomes:

After studying this course, students will be able to:

1. Demonstrate knowledge & examine use of basic statistics and probability.
2. Characterize probability models using probability mass (density) functions &cumulative distribution functions.
3. Developing discrete & continuous probability distributions and its applications.
4. Demonstrate knowledge and be able to apply sampling distributions and limit theorems.
5. Understand methods of inference and estimation and apply this for various statistical hypothesis testing.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Conduct Investigations of Complex Problems
4. Life-Long Learning

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. Probability and Statistics for Engineers and Scientists, 8th Edition, Walpole, Myers, Myers and Ye, Pearson Education, 2007.

Reference Books:

1.Probability& Statistics with Reliability, Queuing and Computer Applications by Kishor S. Trivedi ,2nd Edition, Wiley India, 2014.

2.Probability, Statistics and Random Processes by T.Veerarajan, Tata McGraw Hill.

INTRODUCTION TO WEB DEVELOPMENT

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS363**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Understand the importance of the web as a medium of communication.
- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
- Explain graphic design principles that relate to web design and learn how to implement these theories into practice.
- Develop skills in analysing the usability of a web site.
- Understand and use of language of the web: HTML, CSS, JavaScript, Perl and CGI.

Module -1	**Teaching Hours**	**RBT Levels**
How the Web Works: Definitions and History, Internet Protocols, The Client-Server Model, Where is the Internet, Domain Name System , Uniform Resource Locators,Hypertext Transfer Protocol,Web Servers, What is HTML and Where Did It Come from,HTMLSyntax,SemanticMarkup, Structure of HTML Documents,Quick Tour of HTML Elements,HTML5 Semantic Structure Elements.	**08Hours**	**L1, L2**
Module -2		
What is CSS? , CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling,HTML Tables and Forms: Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats.	**08Hours**	**L1, L2, L6**
Module - 3		
Advanced CSSLayout: Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks. JavaScript-Client-Side Scripting: What is JavaScript and What can it Do?, JavaScript Design Principles, Where Does JavaScript Go? Syntax, JavaScript Objects, The Document Object Model(DOM), JavaScript Events , Forms.	**08Hours**	**L1, L2, L3,L6**
Module-4		
Programming in Perl 5-Why Perl? On-line Documentation, The Basic Perl Program, Scalars, Arrays, Hashes, Control Structures, Processing Text, Regular Expressions, Using Files, Subroutines, Bits and Pieces.	**08Hours**	**L1, L2, L3, L4**
Module-5		

CGI Scripting: What is CGI? Developing CGI Applications, Processing CGI, Introduction to CGI.pm, CGI.pm Methods, Creating MTL pages Dynamically, Using CGI.pm- An Example, Adding Robustness, Carp, Cookies, Uploading Files, Tracking Users With Hidden Data, Creating and Manipulating Images.	**08Hours**	**L1, L2, L3, L6**

Course outcomes:

After studying this course, students will be able to:

1. Interpret internet related technologies
2. Understand the various steps in designing a creative and dynamic website.
3. Develop a website systematically.
4. Write HTML, CSS, JavaScript, Perl and CGI codes.
5. Design dynamic and interactive web pages by embedding Java Script code in HTML.
6. Create good, effective and customized websites.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Design/Development of Solutions
3. Modern Tool Usage
4. The Engineer and Society

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. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", Pearson, 2015.
2. Chris Bates, "Web Programming", 3rd Edition, Wiley, 2006.

Reference Books:

1. Thomas A. Powell, "The Complete Reference HTML& CSS", 5th Edition, McGraw Hill.
2. Brian D Foy," Mastering Perl", O'Reilly Media

DESIGN OF PROGRAMMING WITH LOGIC

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS364**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Understand problem solving skills without imposing the overhead of traditional programming notations and tools.
- Understand design process in problem solving that leads problem statements to well organized solutions.
- Understand programming language details, algorithmic minutiae, and specific application domains.
- Emphasize on algorithmic minutiae, and specific application domain.

Module -1	**Teaching Hours**	**RBT Levels**
Processing of simple forms of Data, Students, teachers & Computers, Numbers, expressions, simple programs, programs are function plus variable definitions, conditional expressions and functions, symbolic information's, compounds data, verities of data.	**08Hours**	**L1, L2**
Module -2		
Syntax and semantics, Processing arbitrarily large data, lists, more on processing lists, natural numbers, composing functions. [Text Book 1]. Problem solving concepts and Planning your solution [chapter2 &3 of Text book 2]	**08Hours**	**L1, L2, L4**
Module - 3		
More on processing arbitrarily large data, self-referential data definitions, mutually referential, development through iterative refinement, processing two complex pieces of data.[text Book 1]. Introduction to Programming structure and Problem solving with sequential logic structure [Text Book 2]	**08Hours**	**L1, L2, L4**
Module-4		
Local definitions and lexical scope, abstracting designs, similarities in definitions, functions are values, designing abstraction from examples & with first class functions, mathematical examples .[Text book 1], Problem solving with decision [Text Book 2]	**08Hours**	**L1, L2, L4**
Module-5		
Generative recursion, designing algorithms, variations on a theme, Algorithm that backtrack, cost of computing and vectors, the loss of knowledge, designing accumulator style functions, Nature of intact numbers, overflow, underflow, DrScheme's numbers.	**08Hours**	**L1, L2, L4**
Course outcomes:		

After studying this course, students will be able to:

- Develop a complete algorithm for a given problem
- Develop novel programming environment.
- Analyze the problem domain clearly.
- Interpret functions and their advantages and roles
- Explain recursion, backtrack, and styles

Graduate Attributes (as per NBA)

1. Problem Analysis
2. Design/Development of Solutions
3. Conduct Investigations of Complex Problems

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. How to Design Programs , Matthias Felleisen, Robert Bruce Findler, Mathew Flatt, Shriramkrishnamurthi, PHI, ISBN-978-81-203-2461-9,Eastern Economy edition
2. Problem Solving and Programming Concepts, 9th Edition, Maureen Sprankle, Jim Habbard, Pearson, ISBN 978-93-325-1884-1

Reference Books: NIL

ANALOG AND DIGITAL ELECTRONICS LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL37**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 02

Course objectives: This laboratory course enable students to get practical experience in design, assembly and evaluation/testing of

- Analog components and circuits including Operational Amplifier, Timer, etc.
- Combinational logic circuits.
- Flip - Flops and their operations
- Counters and Registers using Flip-flops.
- Synchronous and Asynchronous Sequential Circuits.
- A/D and D/A Converters

Descriptions (if any)

Any simulation package like MultiSim / P-spice /Equivalent software may be used.
Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-upon analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-upon Logic design components, pin diagram(if any), 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.

Laboratory Experiments: **RBT Levels: L5, L6**

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 Astablemultivibrator circuit using 555 timer for a given frequency and duty cycle.

Continued: **RBT Levels: L5, L6**

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

5. a)Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
   b) Design and develop the Verilog /VHDL code for an 8:1 multiplexer. Simulate and verify its working.

6. a) Design and implement code converter I)Binary to Gray II) Gray to Binary Code using basic gates.

7. 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.

8. 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. Simulate and verify its working.

9. a) Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
   b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.

10. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447).

11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).

Study experiment

12. To study 4-bitALU using IC-74181.

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Use various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- design and demonstrate various combinational logic circuits.

- design and demonstrate various types of counters and Registers using Flip-flops
- Use simulation package to design circuits.
- Understand the working and implementation of ALU.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (1 to 11 nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:
   a) For questions having part a only- Procedure + Conduction + Viva:**20 + 50 +10 =80 Marks**
   b) For questions having part a and b
      Part a- Procedure + Conduction + Viva:1**0 + 35 +05= 50 Marks**
      Part b- Procedure + Conduction + Viva:1**0 + 15 +05= 30 Marks**
5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

DATA STRUCTURES WITH C LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL38**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 02

Course objectives: This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Asymptotic performance of algorithms.
- Linear data structures and their applications such as Stacks, Queues and Lists
- Non-Linear Data Structures and their Applications such as Trees and Graphs
- Sorting and Searching Algorithms

Descriptions (if any)

Implement all the experiments in C Language under Linux / Windows environment.

Laboratory Experiments: RBT Levels: L3, L4, L5, L6

1. Design, Develop and Implement a menu driven Program in C for the following **Array** operations
   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**
   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**)
   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
   a. Evaluation of **Suffix expression** with single digit operands and operators: **+, -, *, /, %, ^**
   b. Solving **Tower of Hanoi** problem with **n** disks
6. 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**)
   a. Insert an Element on to Circular QUEUE

b. Delete an Element from Circular QUEUE
c. Demonstrate ***Overflow*** and ***Underflow*** situations on Circular QUEUE
d. Display the status of Circular QUEUE
e. Exit

Support the program with appropriate functions for each of the above operations

Continued: RBT Levels: L3, L4, L5, L6

7. 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: ***USN, Name, Branch, Sem, PhNo***
   a. Create a **SLL** of **N** Students Data by using ***front insertion***.
   b. Display the status of **SLL** and count the number of nodes in it
   c. Perform Insertion and Deletion at End of **SLL**
   d. Perform Insertion and Deletion at Front of **SLL**
   e. Demonstrate how this **SLL** can be used as **STACK** and **QUEUE**
   f. Exit

8. 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: ***SSN, Name, Dept, Designation, Sal, PhNo***
   a. Create a **DLL** of **N** Employees Data by using ***end insertion***.
   b. Display the status of **DLL** and count the number of nodes in it
   c. Perform Insertion and Deletion at End of **DLL**
   d. Perform Insertion and Deletion at Front of **DLL**
   e. Demonstrate how this **DLL** can be used as **Double Ended Queue**
   f. Exit
9. Design, Develop and Implement a Program in C for the following operations on **Singly Circular Linked List (SCLL)** with header nodes
   a. Represent and Evaluate a Polynomial **P(x,y,z) = $6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$**
   b. Find the sum of two polynomials **POLY1(x,y,z)** and **POLY2(x,y,z)** and store the result in **POLYSUM(x,y,z)**

   Support the program with appropriate functions for each of the above operations
10. Design, Develop and Implement a menu driven Program in C for the following operations on **Binary Search Tree (BST)** of Integers
    a. Create a BST of **N** Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
    b. Traverse the BST in Inorder, Preorder and Post Order
    c. Search the BST for a given element (**KEY**) and report the appropriate message
    d. Delete an element(**ELEM**) from BST
    e. Exit

11. Design, Develop and Implement a Program in C for the following operations on **Graph(G)** of Cities
    a. Create a Graph of **N** cities using Adjacency Matrix.
    b. Print all the nodes **reachable** from a given starting node in a digraph using **BFS** method
    c. Check whether a given graph is **connected** or not using **DFS** method.

12. 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 →L** as H(**K**)=**K** 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**.

Course outcomes:

On the completion of this laboratory course, the students will be able to:

- Analyze and Compare various linear and non-linear data structures
- Code, debug and demonstrate the working nature of different types of data structures and their applications
- Implement, analyze and evaluate the searching and sorting algorithms
- Choose the appropriate data structure for solving real world problems

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

Conduction of Practical Examination:

1. All laboratory experiments (**TWELVE** nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva:**20 + 50 +10 (80)**
5. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**

ENGINEERING MATHEMATICS-IV

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15MAT41**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Understand mathematics fundamentals necessary to formulate, solve and analyze engineering problems
- Understand and apply Numerical methods to solve ordinary differential equations
- Understand and use Finite difference method to solve partial differential equations
- Perform Complex analysis
- Understand and use of Sampling theory
- Understand and apply Joint probability distribution and stochastic process

Module -1	**Teaching Hours**	**RBT Levels**
Numerical Methods: Numerical solution of ordinary differential equations of first order and first degree, Picard's method, 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). Numerical solution of simultaneous first order ordinary differential equations, Picard's method, Runge-Kutta method of fourth order	**10Hours**	**L1, L2, L3, L4**
Module -2		
Numerical Methods :Numerical solution of second order ordinary differential equations, Picard's method, Runge-Kutta method and Milne's method. **Special Functions:** Bessel's functions- basic properties, recurrence relations, orthogonality and generating functions. Legendre's functions - Legendre's polynomial, Rodrigue's formula, problems.	**10 Hours**	**L1, L2, L3, L4**
Module - 3		
Complex Variables: Function of a complex variable, limits, continuity, differentiability,. Analytic functions-Cauchy-Riemann equations in Cartesian and polar forms. Properties and construction of analytic functions. Complex line integrals-Cauchy's theorem and Cauchy's integral formula, Residue, poles, Cauchy's Residue theorem with proof and problems. **Transformations:** Conformal transformations, discussion of transformations:$w = z^2, w = e^z, w = z + (a^2/z)$ and bilinear transformations.	**10 Hours**	**L1, L2, L3, L4**
Module-4		

Probability Distributions: Random variables(discrete and continuous), probability functions. Poisson distributions **,** geometric distribution, uniform distribution, Exponential and normal distributions, Problems. **Joint probability distribution:** Joint Probability distribution for two variables, expectation, covariance, correlation coefficient.	**10 Hours**	**L1, L2, L3, L4**
Module-5		
Sampling Theory: Sampling, Sampling distributions, standard error, test of hypothesis for means and proportions, confidence limits for means, student's t-distribution, Chi-square distribution as a test of goodness of fit. **Stochastic process:** Stochastic process, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains, higher transition probability.	**10 Hours**	**L1, L2, L3, L4**

Course outcomes:

After studying this course, the students will be able to

- Use appropriate numerical methods to solve first and second order ordinary differential equations.
- Use Bessel's and Legendre's function which often arises when a problem possesses axial and spherical symmetry, such as in quantum mechanics, electromagnetic theory, hydrodynamics and heat conduction.
- State and prove Cauchy's theorem and its consequences including Cauchy's integral formula, compute residues and apply the residue theorem to evaluate integrals.
- Analyze, interpret, and evaluate scientific hypotheses and theories using rigorous statistical methods .

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Life-Long Learning

Conduct Investigations of Complex Problems

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. B.V.Ramana "Higher Engineering M athematics" Tata McGraw-Hill, 2006
2. B. S. Grewal," Higher Engineering Mathematics", Khanna publishers, 42nd edition, 2013

Reference Books:

1. N P Bali and Manish Goyal, "A text book of Engineering mathematics" , Laxmi publications, latest edition.
2. Kreyszig, "Advanced Engineering Mathematics " - 9th edition, Wiley, 2013
3. H. K Dass and Er. RajnishVerma ,"Higher Engineering Mathematics", S. Chand, 1st ed, 2011

SOFTWARE ENGINEERING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS42**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Understand software engineering principles in building large programs
- Analyse ethical and professional issues and to explain why they are of concern to software engineers
- Understand the process of requirements gathering and their validation
- Study the System models and design patterns
- Discuss the distinctions between validation testing and defect testing
- Understand software quality parameters

Module -1	**Teaching Hours**	**RBT Levels**
Introduction: Professional Software Development, Software Engineering Ethics. Case Studies.**Software Processes**: Models. Process activities. Coping with Change. The Rational Unified Process.	**10Hours**	**L1, L2**
Module -2		
Agile Software Development: Agile methods . Plan-driven and agile development. Extreme programming. Agile project management. Scaling agile methods. **Requirements Engineering**: Functional and non-functional requirements .The software Requirements Document. Requirements Specification . Requirements Engineering Processes. Requirements Elicitation and Analysis. Requirements validation. Requirements Management.	**10 Hours**	**L2, L3, L4**
Module - 3		
System Models: Context models. Interaction models. Structural models. Behavioural models. Model-driven engineering. **Design and Implementation**: Object-oriented design using the UML. Design patterns. Implementation issues. Open source development	**10 Hours**	**L2, L3, L4, L5**
Module-4		
Software Testing: Development testing, Test-driven development, Release testing, User testing. **Software Evolution**: Evolution processes .Program evolution dynamics. Software maintenance. Legacy system management	**10 Hours**	**L2, L3, L4, L5**
Module-5		

Project Planning: Software pricing. Plan-driven development. Project scheduling. Agile planning. Estimation techniques. **Quality management** : Software quality. Software standards. Reviews and inspections. Software measurement and metrics.	**10 Hours**	**L2, L3, L4, L5**

Course outcomes:

After studying this course, the students will be able to

- Design a system, component, or process to meet desired needs within realistic constraints.
- Asses professional and ethical responsibility
- function on multi-disciplinary teams
- use the techniques, skills, and modern engineering tools necessary for engineering practice
- Analyse, design, verify, validate, implement, apply, and maintain software systems.

Graduate Attributes (as per NBA)

1. Project Management and Finance
2. Conduct Investigations of Complex Problems
3. Modern Tool Usage
4. Ethics

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. **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)

Reference Books:

1. Roger.S.Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill
2. **PankajJalote**: An Integrated Approach to Software Engineering, Wiley India

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS43**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Understand and analyse the asymptotic performance of algorithms.
- Demonstrate the familiarity with major kinds of algorithms.
- Understand and use of Divide and Conquer method, Greedy Method, Dynamic programming and Backtracking methods in solving problems
- Judge suitable algorithmic design paradigms for real life problems
- Synthesize efficient algorithms in common engineering design situations

Module -1	**Teaching Hours**	**RBT Levels**
Introduction: What is an Algorithm?, Algorithm Specification, Performance Analysis: Space complexity, Time complexity. Asymptotic Notations: Big-Oh notation, Omega notation, Theta notation and Little-oh notation, *Important Problem Types:*Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. *Fundamental Data Structures:* Stacks, Queues, Graphs, Trees, Sets and Dictionaries.	**10Hours**	**L1, L2,**
Module -2		
Divide and Conquer: General method, Binary search, Recurrence equation for Divide and Conquer, Finding the maximum and minimum, Quick sort, Merge sort, Strassen's matrix multiplication, Convex Hull, Closest-pair problem, Advantages and Disadvantages of Divide and Conquer. *Decrease and Conquer Approach*: Topological Sort	**10 Hours**	**L2, L3, L4, L5, L6**
Module - 3		
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.*	**10 Hours**	**L2, L3, L4, L5, L6**
Module-4		
Dynamic Programming: General method, Multistage Graphs, *All Pairs Shortest Paths:* Floyd's Algorithm, Optimal Binary Search Trees, 0/1-Knapsack problem, Bellman-Ford Algorithm, Travelling Sales Person problem, Reliability design.	**10 Hours**	**L2, L3, L4, L5, L6**
Module-5		

Backtracking: General method, N-Queens problem, Sum of subsets problem, Graph colouring, Hamiltonian cycles. **Branch and Bound:** General method, Travelling Sales Person problem,0/1 knapsack problem: LC Branch and Bound solution, FIFO Branch and Bound solution. **NP-Hard and NP-Complete problems:** Basic concepts, non deterministic algorithms, NP - Hard and NP-Complete Classes.	**10 Hours**	**L2, L3, L4, L5, L6**

Course outcomes:

After studying this course, the students will be able to

- ✓ Asses the correctness of algorithms using inductive proofs and loop invariants.
- ✓ Analyse and Compare the asymptotic behaviors of functions obtained by elementary composition of polynomials, exponentials, and logarithmic functions.
- ✓ Describe the relative merits of worst-, average-, and best-case analysis.
- ✓ Describe, apply and analyse the different algorithm design techniques: divide-and-conquer, dynamic programming, greedy paradigm, graph algorithms and their analysis.
- ✓ Judge the applicability of appropriate method for solving real world problems

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Conduct Investigations of Complex Problems
5. Life-Long Learning

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. Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd edition, 2014, Universities Press

2. Introduction to the Design and Analysis of Algorithms, AnanyLevitin:, 3rd Edition, 2012, Pearson

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)

MICROPROCESSORS AND ARM PROCESSORS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - IV

Subject Code	**15CS44**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course Objectives: To make/enable students to

- Familiar with importance and applications of Microprocessors, Microcontrollers, ARM processors
- Understand architecture of 8086 microprocessor and ARM processor
- Understand instruction set of 8086 and ARM processor and write 8086 ALPs
- Write hybrid (assembly & C) program for ARM processor
- Interface peripheral devices like Keyboard, LCD, sensors and stepper motor with ARMprocessor

Module -1	**Teaching Hours**	**RBT Levels**
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, Full Segment Definition, Flowcharts and Pseudo code. **Text book 1: Ch 1: 1.1 to 1.7, Ch 2: 2.1 to 2.7**	**10 Hours**	**L1,L2**
Module -2		
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. **Text book 1: Appendix B, Ch 3: 3.1 to 3.5, Ch 4: 4.1 , 4.2 Chapter 14: 14.1 and 14.2**	**10 Hours**	**L2, L3**
Module - 3		

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 address MAP of x86 PC's, programming and interfacing the 8255.	**10 Hours**	**L2,L3, L4.L6**
Module-4		
Introduction to ARM: RISC and CISC Architectures, T**he ARM Architecture**: The Acorn RISC Machine, The ARM programmer's model: General purpose registers, CPSR, SPSR, ARM memory map, data format, load and store architecture, ARM development tools. ARM Assembly language Programming **Text book 2: Ch 1: 1.5 and 1.6, Ch 2:2.1,2.2,2.3,2.4, Ch 3: 3.1 to 3.5**	**10 Hours**	**L1, L2, L3**
Module-5		
Cortex-M3 series block diagram, registers, instruction set, and addressing modes,Embedded CProgramming for ARM7, C Programs for ARM microprocessor in KEIL, Interfacing ARM7TDMI/ Cortex-M3 to LCD, Keyboard, DAC, sensors and Stepper motor. **Ref book 4: Ch 1: 1.1,1.2,1.3,1.5 Text book 3: Ch 5.**	**10 Hours**	**L1, L2, L3,L6**
Course outcomes:		
After studying this course, the students will be able to • Differentiate microprocessors and microcontrollers • Design and develop 8086 assembly language code to solve problems • Gain the knowledge to interface various devices to ARM processor • Design and developing interrupts routines for interfacing devices		
Graduate Attributes (as per NBA) 1. Engineering Knowledge 2. Modern Tool Usage 3. Design/Development of Solutions 4. Conduct Investigations of Complex Problems		
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 GillispieMazidi, Danny Causey, The x86 PC Assembly Language Design and Interfacing, 5th Edition, Pearson, 2013.
2. ARM System-on-Chip Architecture, Steve Furber, Second Edition, Pearson, 2015
3. ARM System Developer's Guide, Designing and Optimizing Software, Andrew N. Sloss,Dominic Symes, Chris W wight, Elsevier, 2014

Reference Books:

1. Barry B Brey: The Intel Microprocessors, 8th Edition, Pearson Education, 2009.
2. Douglas V. Hall: Microprocessors and Interfacing, Revised 2nd Edition, TMH, 2006.
3. K. Udaya Kumar & B.S. Umashankar : Advanced Microprocessors & IBM-PC Assembly Language Programming, TMH 2003.
4. The Definitive Guide to the ARM Cortex-M3, by Joseph Yiu, 2nd Edition , Newnes, 2009

OBJECT ORIENTED PROGRAMMING WITH C++

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS45**	**IA Marks**	**20**
Number of Lecture Hours/Week	**04**	**Exam Marks**	**80**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**

CREDITS - 04

Course objectives: This course will enable students to

- Differentiate between object oriented programming and procedure oriented programming.
- Define and Describe Classes, objects, constructors, destructors, inheritance, operator overloading, andPolymorphism, Template and exception handling.
- Develop the skills of designing and developing C++ programs using OOP features.
- Disseminate the importance of Object oriented programming

Module -1	**Teaching Hours**	**RBT Levels**
Overview of C++: The Origins of C++ ,What Is Object-Oriented Programming? ,Encapsulation ,Polymorphism , Inheritance. Some C++ Fundamentals ,A Sample C++ Program ,A Closer Look at the I/O Operators, Declaring Local Variables ,No Default to int, The bool Data Type , Old-Style vs. Modern C++ , The New C++ Headers , Namespaces, Working with an Old Compiler, Introducing C++ Classes, Function Overloading, Operator Overloading ,Inheritance Constructors and Destructors ,The C++ Keywords, The General Form of a C++ Program. **Classes and Objects:** Classes, Structures and Classes Are Related, Unions and Classes Are Related , Anonymous Unions, Friend Functions, Friend Classes, Inline Functions, Defining Inline Functions Within a Class Parameterized Constructors, Constructors with One Parameter: A Special Case Static Class Members ,Static Data Members ,Static Member Functions ,When Constructors and Destructors Are Executed ,The Scope Resolution Operator, Nested Classes, Local Classes, Passing Objects to Functions ,Returning Objects ,Object Assignment.	**10Hours**	**L1,L2, L4**
Module -2		

Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects, Creating Initialized vs. Uninitialized Arrays , Pointers to Objects, Type Checking C++ Pointers , The this Pointer, Pointers to Derived Types , Pointers to Class Members , References, Reference Parameters, Passing References to Objects, Returning References, Independent References, References to Derived Types, Restrictions to References, A Matter of Style , C++'s Dynamic Allocation Operators ,Initializing Allocated Memory ,Allocating Arrays ,Allocating Objects ,The nothrow Alternative ,The Placement Form of new.**Function Overloading, Copy Constructors and Default Arguments:** Function Overloading, Overloading Constructors, Overloading a Constructor to Gain Flexibility, Allowing Both Initialized and Uninitialized Objects Copy Constructors , Finding the Address of an Overloaded Function , The overload Anachronism, Default Function Arguments, Default Arguments vs. Overloading, Function Overloading and Ambiguity.	**10 Hours**	**L2,L3**
Module - 3		
Operator Overloading: Creating a Member Operator Function, Creating Prefix and Postfix Forms, of the Increment and decrement Operators , Overloading the Shorthand Operators, Operator Overloading Restrictions, Operator Overloading Using a Friend Function,Using a Friend to Overload ++ or – –, Friend Operator Functions Add Flexibility , Overloading new and delete, Overloading new and delete for Arrays, Overloading the nothrow Version of new and delete, Overloading Some Special Operators, Overloading [ ] , Overloading ( ), Overloading –>, Overloading the Comma Operator.**Inheritances:** Base-Class Access Control, Inheritance and protected Members, Protected Base-Class Inheritance, Inheriting Multiple Base Classes, Constructors, Destructors, and Inheritance, When Constructors and Destructors Are Executed , Passing Parameters to Base-Class Constructors, Granting Access, Virtual Base Classes.	**10 Hours**	**L2,L3, L4**
Module-4		
Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference, The Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical, Pure Virtual Functions, Abstract Classes, Using Virtual Functions, Early vs. Late Binding. **Templates:** Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template, Using Standard Parameters with Template Functions , Generic Function Restrictions , Applying Generic Functions, A Generic Sort, Compacting an Array, Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes, Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates .**Exception Handling:** Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements, Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions, Rethrowing an Exception, Understanding terminate( ) and unexpected( ), Setting the Terminate and Unexpected Handlers, The uncaught_exception( ) Function, The exception and bad_exception Classes Applying Exception Handling.	**10 Hours**	**L2,L3, L6**
Module-5		

C++ I/O System Basics : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O, Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags,An Overloaded Form of setf( ), Examining the Formatting Flags, Setting All Flags, Using width( ), precision( ), and fill( ), Using Manipulators to Format I/O, Overloading << and >>, Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions **C++ File I/O:** fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files, Unformatted and Binary I/O, Characters vs. Bytes, put( ) and get( ), read( ) and write( ), More get( ) Functions, getline( ) , Detecting EOF, The ignore( ) Function, peek( ) and putback( ), flush( ), Random Access, Obtaining the Current File Position , I/O Status, Customized I/O and Files.	**10 Hours**	**L2,L3 L5,L6**

Course outcomes:

After studying this course, the students will be able to

- Differentiate object oriented programming and procedural programming.
- Understand and use the concepts of Object Oriented Paradigm
- Design and develop C++ programs using OOPs features.
- Apply the knowledge gained in the
  a. Understanding of Java and other object oriented programming languages.
  b. Design and Development of wide range of object oriented software packages.
- Acquire competency in using OOPs in different platforms.
- Understand the importance of life-long learning in the field of OOPs.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Design/Development of Solutions
3. Modern Tool Usage
4. Conduct Investigations of Complex Problems

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 Book

1. Herbert Schildt: C++ The Complete Reference, 4th Edition, Tata McGraw Hill, 2014. (Listed topics only from Chapters 11,12,13,14, 15, 16, 17, 18, 19, 20, and21)

Reference Books:

1. Stanley B.Lippmann, JoseeLajore: C++Primer, 4th Edition, Addison Wesley.
2. Joyce Farrell, Object-Oriented Programming Using C++, 4th edition, Cengage Learning.
3. K R Venugopal, RajkumarBuyya,TRavishanker:Mastering C++,Tata McGraw Hill.

INTRODUCTION TO CYBER SECURITY AND CYBER LAW

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS461**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Understand the area of cybercrime and forensics.
- Understand the motive and causes for cybercrime, detection and handling.
- Study the areas affected by cybercrime and investigation.
- Understand the tools used in cyber forensic
- Know Legal Perspectives in cyber security

Module -1	**Teaching Hours**	**RBT Levels**
Introduction to Cybercrime: Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Who are Cybercriminals?, Classifications of Cybercrimes, Cybercrime: The Legal Perspectives, Cybercrimes: An Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes, Cybercrime Era: Survival Mantra for the Netizens. **Cyber offenses: How Criminals Plan Them:** How Criminals Plan the Attacks, Social Engineering, Cyberstalking, Cybercafe and Cybercrimes, Botnets: The Fuel for Cybercrime, Attack Vector, Cloud Computing.	**08 Hours**	**L1, L2**
Module -2		
Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security,Attacks on Mobile/Cell Phones,Mobile Devices: Security Implications for organizations, Organizational Measures for Handling Mobile,Organizational Security Policies and Measures in Mobile Computing Era, Laptops	**08 Hours**	**L1, L2**
Module – 3		
Tools and Methods Used in Cybercrime: Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks **Phishing and Identity Theft:** Introduction, Phishing, Identity Theft (ID Theft)	**08 Hours**	**L1, L2**
0Module-4		

Understanding Computer Forensics: Introduction, Historical Background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber forensics and Digital Evidence, Forensics Analysis of E-Mail, Digital Forensics Life Cycle, Chain of Custody Concept, Network Forensics, Approaching a Computer Forensics Investigation, Setting up a Computer Forensics Laboratory: Understanding the Requirements, Computer Forensics and Steganography, Relevance of the OSI 7 Layer Model to Computer Forensics, Forensics and Social Networking Sites: The Security/Privacy Threats, Computer Forensics from Compliance Perspective, Challenges in Computer Forensics, Special Tools and Techniques, Forensics Auditing, Antiforensics.	**08 Hours**	**L1, L2**
Module-5		
Introduction to Security Policies and Cyber Laws: Need for An Information Security Policy, Information Security Standards – ISO, Introducing Various Security Policies and Their Review Process, Introduction to Indian Cyber Law, Objective and Scope of the it Act, 2000, Intellectual Property Issues, Overview of Intellectual - Property - Related Legislation in India, Patent, Copyright, Law Related to Semiconductor Layout and Design, Software License.	**08Hours**	**L1, L2**
Course outcomes:		
After studying this course, the students will be able to • Acquire knowledge on the cybersecurity cybercrime and forensics. • Explain cybercrime on various mobile and wireless devices • Interpret computer forensics • Understand legal issues in cybercrime. • Use of Tools and methods in cybercrime and security.		
Graduate Attributes (as per NBA) 1. Modern Tool Usage 2. The Engineer and Society 3. Ethics		
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.** SunitBelapure and Nina Godbole, "Cyber Security: Understanding Cyber Crimes, Computer Forensics And Legal Perspectives", Wiley India Pvt Ltd, ISBN: 8126521791, Publish Date 2011 **2.** Dr. Surya PrakashTripathi, RitendraGoyal, Praveen Kumar Shukla, KLSI. "Introduction to information security and cyber laws". Dreamtech Press. ISBN 13: 9789351194736		
Reference Books:		

1. Thomas J. Mowbray, “Cybersecurity: Managing Systems, Conducting Testing, and Investigating Intrusions”, Copyright © 2014 by John Wiley & Sons, Inc, ISBN: 978 -1-118 -84965 -1

2. James Graham, Ryan Olson, Rick Howard, “Cyber Security Essentials”, CRC Press, 15-Dec-2010

GRAPH THEORY AND ITS APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS462**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Understand and apply graphs as a powerful modelling tool that can be used to solve practical problems in various fields.
- Illustrate the main concepts of graph theory, graph representations and the basic classes of graphs.
- Identify induced sub graphs, cliques, matchings, covers in graphs
- Solve famous graph associated problems.
- Use of Optimization and matching algorithms

Module -1	**Teaching Hours**	**RBT Levels**
Introduction to Graph Theory: Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits	**08Hours**	**L1, L2**
Module -2		
Introduction to Graph Theory *contd*.: Planar Graphs, Hamilton Paths and Cycles, Graph Colouring, and Chromatic Polynomials	**08 Hours**	**L2, L3**
Module - 3		
Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes	**08 Hours**	**L1,L2, L3**
Module-4		
Optimization and Matching: Dijkstra's Shortest Path Algorithm, Minimal Spanning Trees – The algorithms of Kruskal and Prim, Transport Networks – Max-flow, Min-cut Theorem, Matching Theory.	**08 Hours**	**L2,L3, L4**
Module-5		
Graph Algorithms: Computer representation of Graphs, Basic algorithms: Spanning Tree, Set of Fundamental circuits, cut-vertices and separability, planarity testing	**08 Hours**	**L2,L3, L4**

Course outcomes:

After studying this course, the students will be able to

- Solve problems using basic graph theory
- Identify induced sub graphs, cliques, matching, covers in graphs

- Determine whether graphs are Hamiltonian and/or Eulerian
- Solve problems involving vertex and edge connectivity, planarity and crossing numbers
- Solve problems involving vertex and edge coloring
- Model real world problems using graph theory

Graduate Attributes (as per NBA)

1. Design/Development of Solutions
2. Modern Tool Usage
3. Problem Analysis

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.Graph theory with Applications to Engineering and computer Science, NarasinghDeo, Prentice Hall

Reference Books:

1. D.S. Chandrasekharaiah: Graph Theory and Combinatorics, Prism.
2. Chartrand Zhang: Introduction to Graph Theory, TMH
3. GeirAgnarsson& Raymond Geenlaw: Graph Theory Modeling, Applications, and Algorithms, Pearson Education.

PYTHON PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS463**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Learn Various Paradigms of Python Programming.
- Use GUI Programming using Tkinter Python's de-facto standard.
- Handle Files, Lists and Dictionaries in Python.
- Explain How to combine data structures and functions available in Python to solve Problems.
- Assess Python as a Programming Language

Module -1	**Teaching Hours**	**RBT Levels**
Introduction to Computers, Programs, and Python Elementary Programming, History of Python, Basic Features of Python ,Mathematical Functions, Strings, and Objects	**08Hours**	**L1,L2**
Module -2		
Creating Python Programs, Selections, Loops, Functions. Programming examples	**08Hours**	**L3,L5, L6**
Module - 3		
Functional programming, Objects and Classes, More on Strings and Special Methods, GUI Programming Using Tkinter, Programming examples	**08Hours**	**L2, L3,L5, L6**
Module-4		
Lists, Multidimensional Lists, Object Oriented Programming, Inheritance and Polymorphism, Programming examples	**08Hours**	**L2,L5, L6**
Module-5		
Files and Exception Handling, Tuples, Sets, and Dictionaries, Recursion, programming examples	**08Hours**	**L2,L5, L6**

Course outcomes:

After studying this course, the students will be able to:

- Explain Python syntax and semantics
- Understand the concepts of Object-Oriented programming as used in Python.
- Demonstrate the fluency in using Python flow control and functions.
- Write Programs using Lists, Dictionaries and handle Files.
- Design and develop GUI Programming using Tkinter

- Build Data Structures using Python

Graduate Attributes (as per NBA)

4. Design/Development of Solutions
5. Modern Tool Usage
6. Problem Analysis

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. Y. Daniel Liang, "Introduction to Programming Using Python", Pearson, ISBN:978-0-13-274718-9, 2013

2.Exploring Python, Timothy A. Budd, Indian edition, McGraw Hill education, ISBN-13: 978-0-07-132122-8

Reference Books:

1. Kenneth A. Lambert , B.L Juneja , "Fundamentals of Python Programming", Cengage Learning,ISBN:978-81-315-2903-4, 2015

2. Charles Dierbach. "Introduction to Computer Science Using Python: A Computational Problem-Solving Focus", Wiley,ISBN:978-81-265-5601-4,2015

3. Allen B.Downey,"Think Python",O'Reilly,First Edition,2012,ISBN:978-93-5023-863-9

PARALLEL PROGRAMMING with OpenMP

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Subject Code	**15CS464**	**IA Marks**	**20**
Number of Lecture Hours/Week	**03**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 03

Course objectives: This course will enable students to

- Understand Parallel Hardware and Parallel Software.
- Develop message-passing parallel programs using MPI
- Design and Develop shared-memory parallel programs using Pthreads
- Design and Develop shared-memory parallel programs using OpenMP.

Module -1	**Teaching Hours**	**RBT Levels**
Motivation and History: Introduction, Modern Scientific Method, Evolution of Supercomputing, Modern parallel Computers, Seeking Concurrency, Data Clustering, Programming Parallel Computers. **Parallel Architectures**: Interconnection Networks, Processor Arrays, Multiprocessors, Multicomputers, Flynn's Taxonomy.	**08Hours**	**L1,L2**
Module -2		
Parallel Hardware and Parallel Software: Some Background, Modifications to the von Neumann Model, Parallel Hardware, Parallel Software, Input and Output, Performance, Parallel Program Design, Writing and running Parallel Programs, Assumptions.	**08 Hours**	**L2,L3, L6**
Module - 3		
Distributed-Memory Programming with MPI: Getting Started, The Trapezoidal Rule in MPI, Dealing with I/O, Collective Communication, MPI Derived Data types, Performance Evaluation of MPI Programs, A Parallel Sorting Algorithm.	**08 Hours**	**L2,L3, L6**
Module-4		
Shared-Memory Programming with Pthreads: Processes, Threads, and Pthreads, Hello, World, Matrix-Vector Multiplication, Critical Sections, Busy-Waiting, Mutexes, Producer-Consumer Synchronization and semaphores, barriers and Condition Variables, Read-Write Locks, Caches, cache Coherence and False Sharing ,Thread Safety.	**08 Hours**	**L2,L3, L6**
Module-5		
Shared-Memory Programming with OpenMP: Getting Started, The Trapezoidal Rule, Scpe of Variables, The Reduction Clause, More About Loops in OpenMp: Sorting, Scheduling Loops, Producers and Consumers, Caches, Cache Coherence, and False Sharing, Thread-Safety.	**08 Hours**	**L2,L3, L6**

Course outcomes:

After studying this course, the students will be able to

- Explain fundamental concepts of parallel architecture and software
- Explain and use models of parallel programming.
- Design and Develop message-passing parallel programs using MPI framework.
- Design and Develop shared-memory parallel programs using Pthreads.
- Design and Develop shared-memory parallel programs using OpenMP.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Modern Tool Usage
4. Conduct Investigations of Complex Problems
5. Design/Development of Solutions

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. M. J. Quinn, "Parallel programming in C with MPI and OpenMP", Tata McGraw Hill,2003.
2. Peter S. Pacheco, "An introduction to parallel programming", Morgan Kaufmann,2011.

Reference Books:

1. B. Chapman, G. Jost, and Ruud van der Pas, "Using OpenMP", MIT Press

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY WITH C++

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - IV

Laboratory Code	**15CSL47**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02p**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 02

Course objectives: This course will enable students to

- Understand and analyze the asymptotic performance of algorithms.
- Give practical exposure to students on various algorithms.
- Demonstrate working nature of algorithms.
- Design and implement various algorithms studied in the theory class and to know the performance

Descriptions (if any)

Design, develop and implement the specified algorithms for the following problems using C++ Language under LINUX /Windows environment.

Experiment Lists: **RBT Levels: L3, L4, L5, L6**

1. A. Create a structure called ***employee*** with the following details as variables within it.

1. Name of the employee
2. Age
3. Designation
4. Salary

Write a C++ program to create array of objects for the structure to access these and print the

name, age, designation and salary with suitable headings.

B. Write a C++ program to create three objects for a class named **print_object** with data members such as *roll_no*&*name*. Create a member function *set_data()* for setting the data values and *print()* member function to print which object has invoked it using '***this***' pointer

2. A. Write a C++ program to define a Class called **STRING** containing two strings and overload *operator*= = for comparing, *operator*+ for Concatenating, *new* and *delete* for allocating and deallocating memory for **STRING** objects.

B. Write a C++ program to create an object of type class **CIRCLE**. Illustrate the constructor, the copy

constructor, and destructor functions associated with a **CIRCLE** object.

Continued...

3. A. Write a C++ program to demonstrate **virtual function** (polymorphism) by creating a base class**polygon** which has virtual function *area()*. Derive two classes **rectangle** and **triangle** from **polygon** and implement *area()* to calculate and return the area of rectangle and triangle respectively.

B. Write a C++ program to explain class template by creating a ***template T*** for a class named **pair** having two data members of **type T** which are read through a constructor and a member function *get-max()* return the greatest of two numbers to main. Note: the value of T depends upon the data type specified during object creation

4. Sort a given set of **N** integer elements using **Quick Sort** method and Compute its time complexity. Run the program for varied values of **N = 10, 20, 30, 40,50,60,…,100**, and record the time taken to sort. Plot a graph of the time taken versus **N** on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using C++ how the divide-and-conquer method works along with its time complexity analysis: worst, average, and best case.

5. Sort a given set of **N** integer elements using **Merge Sort** method and Compute its time complexity. Run the program for varied values of **N = 10, 20, 30, 40,50,60,…,100**, and record the time taken to sort. Plot a graph of the time taken versus **N** on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using C++ how the divide-and-conquer method works along with its time complexity analysis: worst, average, and best case.

6. Implement in C++ the **0/1 Knapsack** problem using i) **Dynamic** Programming ii) **Greedy** methods.

7. From a given vertex in a weighted connected graph, find **shortest paths** to other vertices using **Dijkstra's** algorithm. Write the program in C++ .

8. Find **Minimum Cost Spanning** Tree of a given undirected graph using i) **Kruskal's algorithm ii) Prim's algorithm**. Implement the program in C++ language.

9. a) Design & Implement in C++ to Find a **subset** of a given set **S** = $\{S_1, S_2,.....,S_n\}$ of **n** positive integers whose SUM is equal to a given positive integer D. For example, if S = {1, 2, 5, 6, 8} and D = 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.

b) Design & Implement the presence of **Hamiltonian Cycle** in an undirected Graph G of N

Course outcomes:

After studying this course, the students will be able to

- ✓ Analyze and Compare the asymptotic behaviours of functions obtained by elementary composition of polynomials, exponentials, and logarithmic functions.
- ✓ Design and develop code for different algorithm design techniques: divide-and-conquer, dynamic programming, greedy paradigm, graph algorithms etc

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Modern Tool Usage
4. Conduct Investigations of Complex Problems
5. Design/Development of Solutions

Conduction of Practical Examination:

1. All laboratory experiments (TWELVE nos) are to be included for practical examination.
2. Studentsare allowed to pick one experimentfrom the lot.
3. Strictlyfollow theinstructions as printed on the cover page of answer script for breakup of marks
4. **Marks distribution: Procedure + Conduction + Viva: 20 + 50 + 10 (80)**
5. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**

MICROPROCESSORS and ARM LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2015 -2016)
SEMESTER - III

Laboratory Code	**15CSL48**	**IA Marks**	**20**
Number of Lecture Hours/Week	**01I + 02P**	**Exam Marks**	**80**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS - 02

Course objectives: This course will enable students to

- Write and Execute 8086 ALPs using MASM/TASM
- Design and Develop programs for interfacing LED displays, Keyboards, DAC/ADC, and various other devices using 8086/ARM processor

Descriptions (if any)

Demonstration and Explanation hardware components and Faculty in-charge should explain 8086 architecture, pin diagram in one slot. The second slot, the Faculty in-charge should explain instruction set types/category etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-upon 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-uponInstruction 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 andpractical sessions. Both sessions are evaluated as lab experiments for 20 marks.

Experiment List: **RBT Levels: L3, L4, L5, L6**

- Develop and execute the following programs using 8086 Assembly Language. Any suitable assembler like **MASM/TASM/any equivalent software** may be used.
- Program should have suitable comments.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.

Software Required: Open source ARM Mbed Development platform

KEIL IDE and Proteus for simulation

SOFTWARE(ALP) PROGRAMS: 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. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.
4. Develop an assembly language program to compute **nCr**using recursive procedure. Assume that 'n' and 'r' are non-negative integers.
5. 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.
6. To write and simulate ARM assembly language programs for data transfer, arithmetic and logical operations.
7. To write and simulate C Programs for ARM microprocessor in KEIL

PART B

8. **a.** Design and develop an assembly program to demonstrate BCD Up-Down Counter on the Logic Controller Interface.
   b. 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.
9. 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. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor is it necessary for the student to compute these values).
10. Design and develop an assembly program to drive a Stepper Motor interface and rotate the motor in specified direction (clockwise or counter-clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).
11. Design and develop an assembly language program to
    a. Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).
    b. Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is tobe displayed on the CRO).
12. To interface LCD with ARM microprocessor. Write and execute programs in C language for displaying text messages and numbers on LCD
13. To interface Stepper motor with ARM microprocessor. Write a program to rotate motor in half step and full step mode

Study Experiments:

- Interfacing of temperature sensor with ARM freedom board (or any other ARM microprocessor board) and display temperature on LCD
- To design ARM cortex based automatic number plate recognition system
- To design ARM based power saving system

Course outcomes:

After studying this course, the students will be able to

1. Explain 80x86/ARM instruction sets
2. Design and implement programs written in 80x86/ARM

3. Interface hardware devices to x86/ARM family
4. Assess processors for various kinds of applications.

Graduate Attributes (as per NBA)

1. Engineering Knowledge
2. Problem Analysis
3. Modern Tool Usage
4. Conduct Investigations of Complex Problems
5. Design/Development of Solutions

Conduction of Practical Examination:

1 . All laboratory experiments (***all* 7 + 7**nos) are to be included for practical examination.
2 . Studentsare allowed to pick one experimentfrom each of the lot.
3 . Strictlyfollow theinstructions as printed on the cover page of answer script for breakup of marks
4 . **PART –A**: Procedure + Conduction + Viva: **10 + 25 +05 (40)**
5 . **PART –B**: Procedure + Conduction + Viva: **10 + 25 +05 (40)**
6. **Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.**