NAGARJUNA

College of Engineering & Technology

An Autonomous College under VTU

VISION

Leadership and Excellence in Education.

MISSION

To fulfill the vision by imparting total quality education replete with the philosophy of blending human values and academic professionalism.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

III & IV Semesters

Scheme and Syllabus

With effect from Academic Year

2015 -16

Third Semester B.E. - Scheme

Sl. No	Course Code	Course	Teaching Dept.	L-T-P-S (Hrs/week)	Total Credits	Marks
1	15CSM31	Engineering Mathematics-III (IC)	Mathematics	3-0-2-0	4	100
2	15CST32	Fundamentals of Computation Engineering	CSE	3-0-0-0	3	100
3	15CSI33	Data Structures with C (IC)	CSE	3-0-2-4	5	100
4	15CST34	Analog and Digital Electronics	CSE	3-0-0-0	3	100
5	15CST35	Computer Organization	CSE	3-0-0-0	3	100
6	15CSI36X	Foundation Elective-I (IC)	CSE	2-0-2-0	3	100
7	15CSL37	Analog and Digital Electronics Laboratory	CSE	1-0-2-0	2	100
8	15CSI38	Virtualization Foundations (IC)	CSE	1-0-2-0	2	100
9	15CSH39	Soft Skills Development	CSE	0-2-0-0	1	100
Total				19-2-10-4	26	900

Foundation Elective-I (IC)

Sl. No	Course Code	Course
1	15CSI361	Computer Communication and Networking
2	15CSI362	Creating Interactive and Responsive Web Pages
3	15CSI363	Principles of Programming

IC – Integrated Course **L – Lecture** **T-Tutorials**
P-Practical **S – Self Study**

Fourth Semester B.E. - Scheme

Sl. No	Course Code	Course	Teaching Dept.	L-T-P-S (Hrs/week)	Total Credits	Marks
1	15CSM41	Engineering Mathematics-IV (IC)	Mathematics	3-0-2-0	4	100
2	15CST42	Formal Languages and Automata Theory	CSE	3-0-0-0	3	100
3	15CST43	Design and Analysis of Algorithms	CSE	3-0-0-0	3	100
4	15CSI44	Microprocessors (IC)	CSE	3-0-2-0	4	100
5	15CSI45X	Foundation Elective-II (IC)	CSE	3-0-2-0	4	100
6	15CST46X	Engineering Elective-III	CSE	3-0-0-0	3	100
7	15CSL47	Design and Analysis of Algorithms Laboratory	CSE	1-0-2-0	2	100
8	15CSI48	Cloud Computing Foundations (IC)	CSE	1-0-2-0	2	100
9	15CSH49	Soft Skills Development	CSE	0-2-0-0	1	100
Total				20-2-10-0	26	900

Foundation Elective-II (IC)

Sl. No	Course Code	Course
1	15CSI451	UNIX and Shell Programming
2	15CSI452	Object Oriented Programming with C++
3	15CSI453	Introduction to Programming using Python

Engineering Elective-III

Sl. No	Course Code	Course
1	15CST461	Introduction to Cyber Security and Cyber Laws
2	15CST462	Linear Integrated Circuits
3	15CST463	Control Systems

IC – Integrated Course **L – Lecture** **T-Tutorials**

P-Practical **S – Self Study**

Engineering Mathematics-III (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSM31	3:0:2:0	4	CIE:50 SEE:50	3 Hours	BS

Course Objectives:

This course will enable students to :

- The course is aimed at developing the application of mathematical skills in solving the engineering problems using computers.
- Learn to use the partial differential equations in engineering applications.
- Use of Transforms in the engineering problems.
- Able to find the approximated solutions to engineering problems numerically.

Syllabus

Module - I

Partial Differential Equations: Formation of PDE –Eliminating the Arbitrary constants and arbitrary functions, solutions of non homogenous PDE by direct integration, Method of separation of variables. Applications to PDE –Derivation of one dimensional of wave equation and solution by separation of variables-with specified boundary conditions. Derivation of one dimensional of Heat equation and solution by separation of variables-with specified boundary conditions. **08 Hours**

Module - II

Fourier Series: Periodic functions, Dirchlet's conditions, Euler's Formulae-Fourier series of periodic functions of period 2l and 2π, Half range Fourier series, Practical harmonic analysis. **08 Hours**

Module - III

Numerical Methods-I: Numerical solutions of Algebraic and transcendental equations-Regula Falsi Method and Newton Raphson Method. Finite Differences-Forward, Backward and Central differences, Newton's Forward Newton's Backward and Sterling's interpolation formulae. Lagrange's Interpolation formula (without proof). Numerical Differentiation using Newton's Forward and Backward formulae.

08 Hours

Module - IV

Numerical Methods-II: Numerical Integration-Trapezoidal rule, Simpson's $1/3^{rd}$ and $3/8^{th}$ rule. Numerical solutions of ordinary differential equations of first order

and first degree- Picard's method, Taylor's Series method, Modified Eulers Method, Runge-Kutta Method of 4th order and Milne's Predictor-Corrector Method.

08 Hours

Module - V

Introduction to MATLAB, and its family, Menus and toolbars, Types of windows and types of files, MATLAB Help system, Basic calculations in MATLAB, Basic variables, Functions-Elementary Mathematical, Built in and User defined functions. Array operations, Matrix operations, Loops: for and while loops, condition statements- if-then and if-then-else statements, plotting of graphs, working with scripts and files, approximations and errors using MATLAB, solutions of initial value problems, solutions of system of equations. **08 Hours**

List of MATLAB Experiments

Sl.no	Name of Experiment
01	Basics of MATLAB
02	Basic operations in MATLAB
03	Basic Vector operations
04	Basic Matrix operations
	Solution of Linear Equations
	Determination of Eigen values and Eigen vectors of a Square Matrix
	Solution of Linear Equations for Undetermined and Over determined Cases.
05	Basic Operations on Complex Numbers
06	Plotting of 2D and 3D Curves
07	Polynomial Evaluation and Determination of Roots of a Polynomial
	Determination of Polynomial using Method of Least Square Curve Fitting
	Determination of Polynomial Fit , Analyzing Residuals, Exponential fit and Error Bounds from Given Data
08	Use of Functions
09	Differentiation and Integration
10	Solution of linear differential equations
	Numerical Solutions of Ordinary Differential Equations by Euler's Method
	Numerical Solutions of Ordinary Differential Equations by 4th order Runge Kutta Method

Course Outcomes:

On completion of this course, the students are able to :

- Form a partial differential equations and their solutions.
- Expressing the given functions as infinite series of sine and cosine.
- Find approximated solutions by numerical methods.
- Use the MATLAB to solve the various types engineering problems.

Text Books:

1. Dr. B.S. Grewal: "Higher Engineering Mathematics", (Chapters 10, 17, 18, 22, 23, 28-30), Khanna Publishers, New Delhi, 42nd Edition, 2012, ISBN No: 9788174091956.
2. N.P. Bali and Dr. Manish Goyal: "A Text Book of Engineering Mathematics", (Chapters 10,16,17,20,22,23), Laxmi Publications (P) Ltd., New Delhi, 9th Edition, 2014, ISBN: 9788131808320.
3. Rudrapratab: "Getting started with MATLAB", (Chapters 1-4), Oxford University press, United Kingdom, Indian Edition, 2014 (reprinted).

Reference Books:

1. Erwin Kreyszig: "Advanced Engineering Mathematics", (Chapters 11,12,19), Wiley Pvt. Ltd India, New Delhi, 9th Edition, 2011, ISBN 13: 9788126531356.
2. B.V. Ramana: "Higher Engineering Mathematics", (Chapters 17-21,32), Tata Mc Graw – Hill Publishing company Limited, New Delhi, 2nd Reprint, 2007, ISBN 13: 978-0-07063417-0.
3. S.S. Sastry: "Introductory methods of numerical analysis", (Chapters 2,3,6), PHI Learning Private, Delhi, 5th Edition, 2013, ISBN: 978-81-203-4592-8.
4. Stormy Attaway: "A practical introduction to programming and problem solving", Elsevier, Boston, 2nd Edition.

E-Resources:

1. http://bookboon.com/en/essential-engineering-mathematics-ebook
2. https://www.free-ebooks.net/ebook/essential-engineering-mathematics
3. http://www.zums.ac.ir/ebooks/mathematics/essential-engineering-mathematic.
4. https://archive.org/details/AdvancedEngineeringMathematics10thEdition
5. www.mathworks.com

Fundamentals of Computation Engineering

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST32	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the logical notation of fundamental concepts such as sets, relations and functions.
- Understand the syntax and semantics of propositional and predicate logic.
- Translate statements from a natural language into its symbolic structures in logic.
- Understand the basic concepts of graph theory.
- Learn how to use graphs as a powerful modeling tool to solve practical problems in various fields.
- Get familiarized with modeling of computational methods.

Module - I

Set Theory: Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn Diagrams.

Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic. **06 Hours**

Module - II

Fundamentals of Logic contd.: Logical Implication – Rules of Inference, The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems. **06 Hours**

Module – III

Relations and Functions: Cartesian Products and Relations, Functions –Plain and One-to-One, Onto Functions –Special Functions, The Pigeon-hole Principle, Function Composition and Inverse Functions. Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions. **10 Hours**

Module – IV

Introduction to Graph Theory: Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, , Hamilton Paths and Cycles. Graph Coloring, and Chromatic Polynomials.

Trees: Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes. **10 Hours**

Module – V

Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, combinations – The Binomial Theorem, Combinations with Repetition.

The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle. **08 Hours**

Course outcomes:

On completion of this course, the students are able to :

- Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations and functions.
- Develop the syntax and semantics of propositional and predicate logic.
- Define, compare and recognize relations and functions and Identify relations and functions with graphs, tables and sets of ordered pairs.
- Apply the abstract concepts of graph theory in modeling and solving non-trivial problems in different fields of study.
- Demonstrate the ability to solve problems using counting techniques and combinatorics.

Text Book:

1. Ralph P. Grimaldi: "Discrete and Combinatorial Mathematics", (Chapters 1-3, 5,7,8,11,12), Pearson Education, 5th Edition, 2006, ISBN: 8177584243, 9788177584240.

Reference Books:

1. Kenneth H. Rosen: "Discrete Mathematics and its Applications", McGraw Hill, New Delhi, 7th Edition, 2010, ISBN : 0073383090.
2. J K Sharma: "Discrete Mathematics", Trinity, India, 4th Edition, 2015, ISBN: 978-93-5138-143-3.
3. D.S. Chandrasekharaiah: "Graph Theory and Combinatorics", Prism, Bengaluru, 4th Edition, 2013, ISBN: 978-81-7286-698-3.
4. Richard A. Brualdi: "Introductory Combinatorics", Pearson Education, India, 4th Edition, 2004, ISBN: 978-0-13-602040-0.

E-Resources:

1. https://www.pearsoned.co.in/grimaldidcm5e

Data Structures with C (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI33	3:0:2:4	5	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the basics of structures and pointers.
- Learn the data structure for a specified application.
- Understand the impact of data structures on the performance of programs.
- Get exposed to various data structure concepts, such as stacks, queues, linear lists, heaps, trees and graphs.

Syllabus

Module - I

Structures and Unions: Introduction, Defining a structure, Declaring structure variables, Accessing structure members, Structure initialization, Size of structures, Copying and comparing structure variables, Array of structures, Arrays within structures, Structures within structures, Structures and functions, Unions.

Pointers: Introduction, Accessing the variable and its address through the pointer, Declaring and initialization of pointer variables, Chain of pointers, Pointer expressions, Pointer increments and Scale factor, Array of pointers, Pointer as function arguments, Functions returning pointers, Pointers to functions, Pointers and arrays, Pointers and character strings, Pointers and structures. Dynamic memory allocation - Introduction, malloc(), calloc(), realloc(). **08 Hours**

Module - II

Stacks: Definition, Representing stacks in C - Implementing the PUSH and POP operation with example and testing for exceptional conditions, Infix, Postfix, and Prefix expressions. Program to evaluate a postfix expression with example, Program to convert an expression from infix to postfix. Recursion: Definition, Recursion in C with examples.

Queues: The Queue and its sequential representation, Implementation of queues in C, Insert and delete operations, Priority queue, Array implementation of a priority queue. **08 Hours**

Module - III

Linked Lists: Introduction, Inserting and removing nodes from a list, Linked list implementation of stacks, queues and priority queues, List operations, get node and

free node. Lists in C - Array implementation of lists and its limitations, Allocating and freeing of dynamic variables, Linked lists using dynamic variables and compare with array implementation of lists, Non-integer and non-homogeneous lists, Circular lists: Primitive operations on circular lists, implement stack and queue as a circular list, Doubly linked lists: Inserting and removing nodes from a list, Addition of long integer using Doubly linked lists. **08 Hours**

Module - IV

Trees: Introduction, Binary trees, Binary tree traversals: preorder, inorder and postorder, Threaded binary trees : insertion and deletion of a node, Heaps: insertion and deletion of a node, Max heap and min heap, Binary search trees: insertion and deletion of a node, searching a key element in binary search tree, Selection trees, Forests, Representation of disjoint sets, Counting binary trees. **08 Hours**

Module - V

Efficient Binary Search Trees: Optimal binary search trees, AVL trees: insertion and deletion of node, Red-black trees: insertion and deletion of a node, Splay trees: insertion and deletion of a node, Graph: Introduction, Abstract data type of graph. **08 Hours**

Laboratory

Design, develop and implement the specified algorithms for the following problems using C/C++ language in LINUX / Windows environment and simulate the programs by using virtual lab.

1. Design, develop and execute a program in C based on the following requirements: An EMPLOYEE class is to contain the following data members and member functions: Data members: Employee_Number (an integer), Employee_Name (a string of characters), Basic_Salary (an integer), All_Allowances (an integer), IT (an integer), Net_Salary (an integer). Member functions: to read the data of an employee, to calculate Net_Salary and to print the values of all the data members. (All_Allowances = 123% of Basic, Income Tax (IT) = 30% of the gross salary (gross salary = Basic_Salary + All_Allowance), Net_Salary = Basic_Salary + All_Allowances – IT).
2. Using circular representation for a polynomial, design, develop and execute a program in C to accept two polynomials, add them and then print the resulting polynomial.
3. Design, develop and execute a program in C to simulate the working of a Stack of integers using an array. Provide the following operations:

   a. Push b. Pop c. Display

4. Design, develop and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations:

   a. Insert b. Delete c. Display

5. Design, develop and execute a program in C to convert a given valid parenthesized infix arithmetic expression to postfix expression and then to print both the expressions. The expression consists of single character operands and the binary operators +, -, * and /.

6. a. Design, develop and execute a program in C to evaluate a valid postfix expression using Stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are +, -, * and /.

   b. Simulation of the above program by using virtual lab.

7. Design, develop and execute a program in C to implement LIST (linked list) with functions to insert an element at the front of the list as well as to delete an element from the front of the list (FIFO), display the contents of the list.

8. a. Design, develop and execute a program in C to implement LIST (linked list) with functions to insert an element at the front of the list as well as to delete an element from the rear of the list or vice versa (LIFO), display the contents of the list.

   b. Simulate a program to use Linked list to do operations on polynomials. (virtual lab)

9. Design, develop and execute a program in C to implement LIST (linked list) with functions to insert an element at the front/rear end of the list and search and delete a key element of the list, if exists and display the contents of the list.

10. Design, develop and execute a program in C to create a max heap of integers by accepting one element at a time and by inserting it immediately in to the heap. Use the array representation for the heap. Display the array at the end of insertion phase.

11. Write a C program to support the following operations on a doubly linked list where each node consists of integers,

    a. Create a doubly linked list by adding each node at the front.

    b. Insert a new node to the left of the node whose key value is read as an input.

    c. Delete the node of a given data, if it is found, otherwise display appropriate message.

    d. Display the contents of the list.

12. Design, develop and execute a program in C to implement BIN_TREE that represents a Binary Tree, with functions to perform inorder, preorder and postorder traversals and demonstrate the traversals.

Course Outcomes:

On completion of this course, the students are able to :

- Describe the concepts of structures and pointers.
- Explain and illustrate the various stack and queue operations.
- Interpret and design the programs using linked lists.
- Implement binary trees, threaded binary trees, binary search trees using C language.
- Implement AVL trees, Red-black trees and Splay trees using C language.

Text Book:

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed : "Fundamentals of Data Structures in C", (Chapters 1-6,9,10), Universities Press-India, 2nd Edition, 2008, ISBN-13: 978-8173716058.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI Learning Private Limited-India, 3rd Edition, 2010, ISBN-13: 978-8120340077.
2. Aaron M. Tenenbaum: "Data Structures using C", Pearson Education, India, 2nd Edition, 2003, ISBN-13: 978-8131702291.

E-Resources:

1. https://www.cs.princeton.edu/
2. https://www.opendatastructures.org/ods-cpp
3. https://www.lib.mdp.ac.in/ebook/DSa
4 https://ww.cs-fundamentals.com/data-structures/introduction-to-data-structures.php
5. https://www.cprogramming.com/algorithms-and-data-structures.html
6. https://www.cprograms.in/

Analog and Digital Electronics

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST34	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand applications of diode as clippers, Limiters and clampers.
- Know the different types of amplifiers constructed from BJTs and MOSFETs.
- Acquire the knowledge about Boolean laws and theorems.
- Understand the various types of data processing circuits.
- Learn the working principles of various Flip-Flops, registers, counters.
- Understand the basic concepts of HDL.

Syllabus

Module - I

Diode Circuits: The ideal diode, The second approximation, The third approximation, DC resistance of a diode, Load lines, Clippers and Limiters, Clampers.

BJT Amplifiers: Emitter-biased amplifier, Small-signal operation, AC Beta, AC resistance of the emitter diode, Two- transistor models, Analyzing an amplifier, Voltage gain. **07 Hours**

Module - II

MOSFETs: The Depletion-mode MOSFET, D-MOSFET curves, Depletion-mode MOSFET amplifiers, The enhancement-mode MOSFET, CMOS, E-MOSFET amplifiers.

OP-AMP's in Waveform conversion and Generation Circuits: Sine to rectangular, Rectangular to triangular, Triangular to pulse conversion circuits, Relaxation oscillator, Generating triangular waves. **09 Hours**

Module - III

Combinational Logic Circuits: Boolean laws and theorems, 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-McClusky method, Introduction to HDL, HDL implementation models.

Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, Encoders, HDL implementation of data processing circuits. **09 Hours**

Module - IV

Flip-Flops: Gated FLIP-FLOPs, Edge-triggered D FLIP-FLOP, Edge-triggered JK FLIP-FLOP, JK Master-slave FLIP-FLOP, 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. **08 Hours**

Module - V

Registers: Applications of shift registers, Register implementation in HDL .

Counters: Asynchronous counters, Decoding gates, Synchronous counters, Changing the counter modulus, Decade counters, Counter design as a synthesis problem.

07 Hours

Course Outcomes:

On completion of this course, the students are able to :

- Use diodes in clippers, limiters and clampers.
- Analyze and design transistor and MOSFET amplifiers in different configurations.
- Apply K-Map and Quine-McClusky methods to simplify the given Boolean expressions.
- Design and implement registers using Flip-Flops.
- Design and develop counters using Flip-Flops.

Text Books :

1. Albert Malvino, David Bates: "Electronic Principles", (Chapters 3,4,8,12,20), TMH, New Delhi, 8th Edition, 2015, ISBN-9780073373881.
2. Donald P Leach, Albert Paul Malvino and Goutam Saha: "Digital Principles and Applications", (Chapters 3,4,8- 10), Tata McGraw Hill, New Delhi, India, 8th Edition, 2014, ISBN: 9789339203402.

Reference Books :

1. Robert L. Boylestad, Louis Nashelsky: "Electronic Devices and Circuit Theory", PHI/Pearson Education, New Delhi, 10th Edition, 2012, ISBN: 9788131764596.
2. David A. Bell: "Electronic Devices and Circuits", Oxford University Press, New Delhi, India, 5th Edition, 2010, ISBN: 9780195693409.
3. M Morris Mano: "Digital Logic and Computer Design", Pearson Education, Prentice Hall, 11th Edition, 2009, ISBN: 9788177584097.
4. R D Sudhaker Samuel: "Illustrative Approach to Logic Design", Sanguine-Pearson, New Delhi India, 2012, ISBN: 9788131765081.

E-Resources:

1. https://www.talkingelectronics.com/download/Malvino_electronic_principles.pdf.
2. https://www.rtna.ac.th/departments/elect/data/EE304/Electronic%20Devices%20and%20Circuit%20Theory.pdf
3. https://www.abebook.com/Digital_Principles_Applications_Seventh_Edition_Albert/4893172428/bd

Computer Organization

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST35	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Become familiar with the basics of computer structure and its performance.
- Learn basics of assembly language programming.
- Understand connections of peripheral devices and interrupts generated by them.
- Understand the basics of data communication by using different types of buses.
- Learn internal structure of memory and CPU.

Syllabus

Module - I

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Performance -Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement, Historical Perspective.

08 Hours

Module - II

Machine Instructions and Programs: Numbers, Arithmetic operations and Characters, 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. **09 Hours**

Module - III

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. **08 Hours**

Module - IV

Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories. **08 Hours**

Module - V

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand multiplications, Fast Multiplications, Integer Division, Floating-point Numbers and Operations. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Describe basic structure and performance of computer.
- Explain basics of assembly language programming.
- Explain connections of peripheral devices, interrupts and DMA operations.
- Use different types of buses for data transmission and design of basic types of memory.
- Construct different types of arithmetic circuits.

Text Book:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: "Computer Organization", (Chapters 1,2,4-6) TMH, Mcgraw-Hill College Division, 5th Edition, 2002. ISBN :9781259005275.

Reference Books:

1. William Stallings: "Computer Organization and Architecture", PHI, Pearson Education, Delhi, 10th Edition, 2016, ISBN: 9780134101613.
2. David. A. Patterson, John L. Hennessy: "Computer Organization and Design – The Hardware / Software Interface", ARM Edition, 5th Edition, Elsevier, 2014, ISBN: 97801240776263.

E-Resources:

1. https://books.google.co.in/books?isbn=0071089004
2. https://books.google.co.in/books?isbn=8177589938
3. https://books.google.co.in/books?isbn=0124078869

Computer Communication and Networking (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI361	2:0:2:0	3	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to :

- Understand the basics of data communication system and network models.
- Identify the need and techniques for digital and analog transmissions.
- Get exposed to different error detection and correction methods.

Syllabus

Module - I

Introduction to Communications: Data Communications, Networks - Physical structure, Categories of Networks, Network Models –The OSI Model, TCP/IP Protocol Suite. **06 Hours**

Module - II

Digital Transmission: Digital to Digital Conversion – Line coding, Line coding schemes (Unipolar, Polar, Bipolar only), Analog to Digital Conversion – PCM Encoder and Decoder, Transmission Modes. **05 Hours**

Module - III

Analog Transmission: Digital to Analog conversion - Introduction, ASK, PSK, FSK, Analog to Analog conversion – Introduction, AM, PM, FM. **05 Hours**

Module - IV

Multiplexing and Spectrum Spreading: Multiplexing - Introduction, FDM, Synchronous TDM, Statistical TDM, , Spread Spectrum – Introduction, FHSS, DSSS. **05 Hours**

Module - V

Error Detection and Correction: Introduction, Block Coding, Cyclic Codes – CRC, Polynomials, Cyclic code encoder using Polynomials, Advantages of cyclic codes. Checksum. **05 Hours**

Laboratory

1. Write a program to convert digital to analog data transmission.
2. Write a program to convert analog to digital data transmission.

3. Write a program for error detecting code using CRC-CCITT (16 bits).
4. Using TCP/IP sockets write a client server program to make the client send the file name and to make the server send back the contents of the requested file if present.

Course Outcomes:

On completion of this course, the students are able to :

- Describe the basics of data communication system and network models.
- Distinguish between different techniques of digital transmissions.
- Compare different methods of analog transmissions.
- Explain various types of multiplexing and spread spectrum mechanisms.
- Solve problems of error detection and correction using Block coding and CRC mechanisms.

Text Book:

1. Behrouz A. Forouzan: "Data Communication and Networking", (Chapters 1,2, 4-6, 10), McGraw Hill Education, New Delhi, India, 5th Edition, Copyright: 2013. Publication Date: February 17, 2012, ISBN: 9781259064753.

Reference Books:

1. William Stallings: "Data and Computer Communication", Pearson Education, Delhi, 8th Edition, 2007, ISBN: 9788131715369.
2. Craig Zacker: "The Complete Reference Networking", McGraw-Hill Education, New Delhi, India, 1st Edition, 2002, ISBN: 13: 978-0070474161.
3. Wayne Tomasi: "Introduction to Data Communications and Networking", Pearson Education, Delhi, 1st Edition, ISBN 13: 978-8131709306.

E-Resources:

1. http://www.mhhe.com/engcs/compsci/forouzan/frontmatter.pdf.
2. http://ebookinga.com/data-communication-and-networking-tata-mcgraw-hil.
3. https://www.goodreads.com/book/show/209441.Introduction_to_Data_Communications_and_Networking.

Creating Interactive and Responsive Web Pages (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI362	2:0:2:0	3	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to :

- Learn the evolution of the World Wide Web and its relevance in today's world.
- Get a clear understanding of the technologies involved in developing a web-site.
- Learn web technology with a focus on creating interactive and responsive web pages.

Syllabus

Module - I

HTML 5 and CSS: Introduction to Hyper Text Markup Language, Key components of HTML document, HTML elements, Headers, Linking, Images, Unordered Lists, and Nested and ordered Lists.

Tables, Divs and forms: HTML Tables and Formatting, HTML Forms, Internal Linking, Creating and Using Images, Maps, Div and span tags. Introduction CSS, CSS selector, positioning, layouts, debugging. **06 Hours**

Module - II

JavaScript: Browser and Document object, scripts and HTML Document, variables, expressions, Data type conversions, decisions and loops, control structure, windows Document object, forms and form handling elements, scripting, event handling.

05 Hours

Module - III

jQuery: Using selectors with jQuery, Manipulating page elements with jQuery, jQuery event model, jQuery and Ajax, jQuery animation and advanced effects, jQuery plugins. **05 Hours**

Module - IV

Bootstrap: Bootstrap Scaffolding, Bootstrap CSS, Bootstrap Layout Components, Bootstrap JavaScript Plugins, Using Bootstrap. **05 Hours**

Module - V

XML: What is XML? What are the differences between HTML and XML, what is the purpose of XML?

AJAX: AJAX Introduction, AJAX XML Http, AJAX Request, AJAX Response, AJAX with Server side. **05 Hours**

Hands on

1. HTML and CSS -

a. Create a HTML page to display the following content <Ensure the format is same as shwn below>; Use HTML Tables UnorderList and OrderList (UL and OL):

1. Development Environment • Eclipse (SpringSource Tool Suite distribution) • Apache Tomcat/VMware® vFabric™ tc Server • Spring Insight • Testing tools	**2. Spring Overview** • Introduction to Spring configuration • Bean life cycle • Simplifying configuration • Integration testing with Spring
3. Getting Started with Spring Web MVC • Spring model-view-controller (MVC) overview • DispatcherServlet • Controller programming model overview • Spring MVC views • Simplifying configuration	**4. Spring MVC Configuration Options** • Spring MVC infrastructure Beans • URL mappings • Handler interceptors and handler adapters • Exception resolvers • Message source

b. Create a HTML page to display testimonials received from customers along with their picture and Name as shown below:

2. Java Script and jQuery

a. jQuery form validations:

Front-End: Develop below form using HTML to create new user:

Field Details:

1. Name <String, Length(16), Mandatory, Validations: Minimum Length: 3, Can accept special characters>
2. Email <String, Length(60), Mandatory, Validations: Should be a valid email id>
3. Password <String, Length(16), Mandatory, Validations: Minimum Length: 5, Can accept special characters>

- Create Buttons - "Create an account" and "Cancel"
- Form should be Scrollable
- For field validations, use jQuery
- Ensure all validations pertaining to Name, Email and Password are taken care. If the use enters incorrect values appropriate error message should be displayed and should allow the user to enter correct data

b. jQuery image slider:

In an HTML page, insert a minimum of 5 images; Ensure inserted images are scrollable.
Hint: To make images scrollable use jQuery image slider or use javascript.

Sample screen shot :

3. Develop below form using HTML to Search and Book Tickets:

Validations :

1. All fields are Mandatory except “Single Lady” Field
2. Onwards date must be less than Return date

4. Bootstrap, AJAX and jQuery:

a. Create a Bootstrap Page that helps maintain Employee Information in an organization.

b. When clicked on the “Add New Employee” button, load a dialog box as shown below

Validations:

1. All fields are Mandatory.
2. On successful submission of the form, the new employee details has to be appended as a last row in the table.
3. When clicked on the Edit icon, a similar pop-up as the "Add new Employee" form has to be displayed with the input fields populated with appropriate values. When the form is submitted in the Edit flow, ensure all the validations are in place. The name of the button in the Edit flow has to be "Update" instead of "Add".
4. When clicked on the Delete icon, a confirmation dialog box has to be displayed with a message "Are you sure, you want to delete this entry?" If the user clicks "Yes", the corresponding row has to be deleted from the table. If the user clicks "No" the table has to remain unaffected.

Course Outcomes:

On completion of this course, the students are able to :

- Develop web layouts with style sheets and web screens in a presentable form.
- Write interactive web pages through form validations and other methods. Use the same in UI development.
- Use the Java Script libraries to accelerate UI development.
- Design and develop responsive and mobile first web pages.
- Develop applications by using synchronous and asynchronous communication over web.

Text Book:

1. Jon Duckett: "Web Design with HTML, CSS, JavaScript and jQuery Set", Wiley, 1st Edition, 2014, ISBN 13: 978-1118907443.

Reference Books:

1. Jake Spurlock: "Bootstrap, Shroff", O'Reilly Media, United States of America, 1st Edition, 2013, ISBN: 978 -1 -4493-4391-0.
2. Bear Bibeault, Yehuda Katz and Aurelio De Rosa: "jQuery in Action", Dreamtech Press, New Delhi, India, 3rd Edition, 2015, ISBN: 978-1617292071.

E-Resources:

1. http://www.w3schools.com/
2. https://learn.jquery.com/
3. https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/JavaScript_basics

Principles of Programming (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI363	2:0:2:0	3	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to :

- Learn data, data types, and basic statements.
- Gain knowledge about call-return architecture and ways of implementing them.
- Understand object-orientation, concurrency, and event handling in programming languages.
- Develop programs in non-procedural programming paradigms.

Syllabus

Module - I

Data, Data Types, and Basic Statements-I: Names, Variables, Binding, Type checking, Scope, Scope rules, Lifetime and garbage collection, Primitive data types, Strings, Array types, Associative arrays, Record types, Union types, Pointers and references. **05 Hours**

Module - II

Data, Data Types, and Basic Statements-II: Arithmetic expressions, Overloaded operators, Type conversions, Relational and boolean expressions, Assignment statements, Mixed-mode assignments, Control structures, Selection, Iterations, Branching, Guarded statements. **05 Hours**

Module - III

Subprograms and Implementations-I: Subprograms, Design issues, Local referencing, Parameter passing, Overloaded methods, Generic methods, Design issues for functions. **05 Hours**

Module - IV

Subprograms and Implementations-II: Semantics of call and return, Implementing simple subprograms, Stack and dynamic local variables, Nested subprograms, Blocks, Dynamic scoping. **05 Hours**

Module - V

Object-Orientation, Concurrency, and Event Handling: Object-orientation, Design issues for OOP languages, Implementation of object-oriented constructs, Concurrency, Exception handling. **06 Hours**

Laboratory

1. Implement all major functions of string.h in single C program using switch case to select specific function from user choice (like strlen, strcat, strcpy, strcmp, strrev).
2. Write a program (WAP) in C to reverse a linked list iterative and recursive.
3. WAP in C to implement iterative Towers of Hanoi.
4. WAP in C++ to count the no's of object of a class with the help of static data member, function and constructor.
5. WAP in C++ to declare a class Time with data members mm for minutes, ss for seconds and hh for Hours.
6. Define a parameterize constructor to assign time to its objects. Add two time objects using member function and assign to third objects. Implement all possible cases of time.

Course Outcomes:

On completion of this course, the students are able to :

- Describe data and data types of programming languages.
- Explain basic statements of programming languages.
- Design subprogram constructs.
- Implement simple subprogram.
- Apply object-oriented, concurrency, and event handling programming constructs.

Text Book:

1. Robert W. Sebesta: "Concepts of Programming Languages", (Chapters 5-10,12, 14), 10th Edition, Addison Wesley, 2012, ISBN: 0-13-607347-6.

Reference Books:

1. R. Kent Dybrig:"The Scheme programming language", 4th Edition, MIT Press, 2009, ISBN: 978-0-262-51298-5.
2. Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009, ISBN: 0 262 15039 5.
3. Michael L. Scott, "Programming Language Pragmatics", 3rd Edition, Morgan Kaufmann, 2009, ISBN-13: 978-0123745149, ISBN-10: 0123745144.

E-Resources:

1. https://www.google.co.in/?gfe_rd=crandamp;ei=1r5PV7SWFtWQ2ASd3q3ADw#q=robert+w.+sebesta+concepts+of+programming+languages
2. https://www.google.co.in/?gfe_rd=crandamp;ei=1r5PV7SWFtWQ2ASd3q3ADw#q=michael+l.+scott+programming+language+pragmatics
3. https://www.google.co.in/?gfe_rd=crandamp;ei=1r5PV7SWFtWQ2ASd3q3ADw#q=programming+language+pragmatics+morgan+kaufmann+pdf

Analog and Digital Electronics Laboratory

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSL37	1-0-2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand and design diode circuits such as clippers and clampers.
- Learn the design of OP-AMP circuits.
- Learn how to implement combinational circuits.
- Know how to design and implement sequential circuits.

Part-A

1. Design and construct a suitable circuit and demonstrate the working of positive clipper, and clamper using diodes.
2. Construct a suitable circuit and determine the frequency response and bandwidth of a CE amplifier.
3. Design and implement relaxation circuit using OP-AMP to generate a rectangular wave from for a given frequency.
4. Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
5. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop and demonstrate its working.
6. Design and implement a ring counter using 4-bit shift register and demonstrate its working.
7. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate its working.

Part-B (Virtual Lab)

1. Demonstrate the working of the positive clipper and clamper circuits.
2. Build the CE amplifier circuit and determine the voltage gain for two different values of supply voltage and for two different values of emitter resistance.
3. Build CMOS inverter using a simulation package and verify its truth table.
4. Develop the Verilog / VHDL code for an 8:1 multiplexer. Simulate and verify its working.
5. Develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify its working.
6. Develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify its working.
7. Develop the Verilog / VHDL code for switched tail counter. Simulate and verify its working.

Course Outcomes:

On completion of this course, the students are able to :

- Analyze and design transistor and MOSFET Amplifiers in different configurations.
- Construct various diode circuits and OP-AMP circuits.
- Realize and verify truth tables of D Flip-Flop and 8:1 Multiplexer.
- Design and implement ring counter.
- Design and implement synchronous and asynchronous counters.

Text Books:

1. Albert Malvino, David Bates: "Electronic Principles", 8th Edition, TMH, New Delhi, 2015, ISBN: 9780073373881.
2. Donald P Leach, Albert Paul Malvino and Goutam Saha: "Digital Principles and Applications", 8th Edition, Tata McGraw Hill, New Delhi, India, 2014, ISBN: 9789339203402.

E-Resources:

1. http://elearning.vtu.ac.in>E-Notes>LD manual.
2. http://Faqexplorer.com

Virtualization Foundations (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI38	1:0:2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Get a strong foundation for mastering computer virtualization.

Syllabus

Module - I

Introduction to Virtualization: Definition of Virtualization, Traditional versus Virtualized System, How Virtualization works, Need for Virtualization, Benefits of Virtualization, Definition of Hypervisor, Type-1 Hypervisor, Type-2 Hypervisor, Comparison of Type1 and Type2 Hypervisors, Types of Virtualization - Server Virtualization, Desktop Virtualization, Application Virtualization, Network Virtualization, Storage Virtualization. **03 Hours**

Module - II

VMware Player: Overview, Introduction to VMware Player, System Requirements, Installation of VMware Player, Enabling VT-x/AMD-v in BIOS, Creating a blank Virtual Machine, Installation of Open Suse 64-bit, Introduction to Virtual Appliance, Demonstration of Virtual Appliance, Introduction to Unity mode, Demonstration of Unity mode. **03 Hours**

Module - III

ESXi: Introduction to ESXi, ESXi Architecture, System Requirements, Installing ESXi, Installing Windows 2008 R2, Installing vSphere Client, Creating a VM on ESXi using vSphere Client, Installing a Guest OS on ESXi. **02 Hours**

Module - IV

Networking and Storage: Introduction to Virtual Networking, Virtual Networking - Components and Concepts, Virtual Networking in ESXi, Introduction to Storage, How Virtual Machines Access Storage, Types of Physical Storage. **03 Hours**

Module - V

VMware Converter: Overview, Installation of VMware Converter, Converting a Physical Computer to a VM, Powering-on the converted VM. **02 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Explain what virtualization is, and get an in-depth understanding of how things work at the ring level.
- Install VMware Player and create virtual machines using VMware Player. In specific, the module will guide on setting up a virtual lab environment using VMware Player.
- Install VMware ESXi, working with the vSphere Client, and managing a host using the vSphere Client.
- Cover the concepts of Virtual Networking and Storage with reference to ESXi.
- Use VMware Converter to convert physical computer into a virtual machine.

Reference Books:

1. Matthew Portnoy: "Virtualization Essentials", Wiley, 2012, ISBN: 978-1-118-17671-9.
2. Nelson Ruest, Danielle Ruest: "Virtualization, A Beginner's Guide", McGraw-Hill Education, 2009, ISBN-13: 978-0071614016.

E-Resources:

1. http://www.vmware.com/files/pdf/gated-vmw-ebook_virtualization-essentials.pdf.

Soft Skills Development

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSH39	0:2:0:0	1	CIE:50 SEE:50	2 Hours	HSS

Course Objectives:

This course will enable students to :

- Improve the communication skills.
- Develop the art of presentation and writing effectively.
- Enhance the technical knowledge.

To improve the communication and presentation skills, every student has to give a seminar on technical topics assigned by the supervisors. Each course coordinator/ faculty members will be assigned with few students to guide and monitor the presentation. The presentation shall be for 15 minutes. A brief report on the seminar has to be submitted by the student to the concerned department after completion of the seminar. The report shall be signed by the supervisor and the Head of the concerned department.

The objective of the seminar is to introduce students to the major constituent of technology that is concerned with critically reading, understanding, summarizing, explaining and presenting existing technical topics. Students have to refer one or more topics that are assigned to them by their supervisors. The idea behind the seminar system is to familiarize student more extensively with the methodology of their chosen subject, allow them to develop presentation skills and also interact with example of practical problems.

Course Outcomes:

On completion of this course, the students are able to :

1. Get rid of stage fear and answer questions from audience.
2. Communicate confidently and fluently.
3. Comprehend and prepare reports effectively.

Engineering Mathematics-IV (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSM41	3:0:2:0	4	CIE:50 SEE:50	3 Hours	BS

Course Objectives:

This course will enable students to :

- The course is aimed at developing the application of mathematical skills in solving statistics and probability problems using computers.
- Applying the sample analysis on the large data.
- Learn to analyze the different Queuing Models.
- Introduction of Statistical Software's.

Syllabus

Module - I

Random Variables: Discrete probability distribution, Continuous probability distribution, Expectation, Variance, Probability generating function, Binomial distribution, Poisson distribution, Exponential distribution and Normal distribution.

08 Hours

Module - II

Joint Probability: Joint probability distribution, discrete and independent random variables, expectation, covariance, correlation coefficient, probability vectors, stochastic matrices, fixed point matrices, regular stochastic matrices, Markov chains, higher transition probabilities, stationary distribution of regular Markov chains and absorbing states. **08 Hours**

Module - III

Sampling: Sampling and inference: Sampling distribution, testing of hypothesis, level of significance, confidence limits. Test of significance of large samples, sampling of variables, central limit theorem, confidence limits for unknown means, students t-distribution.

Queuing Theory: Terminology of Queuing system, Queuing models, Simulation.

08 Hours

Module - IV

Information Theory and Coding: Introduction, Entropy, Joint Entropy and Conditional Entropy, Relative Entropy and Mutual Information, Relationship Between Entropy and Mutual Information, Chain Rules for Entropy, Relative Entropy and Mutual Information, Jensen's Inequality and its Consequences, Log Sum Inequality and its Applications, Data-Processing Inequality, Sufficient Statistics, Fano's Inequality.

08 Hours

Module - V

Introduction to R, Basic Data types, vector operations, matrix construction, lists, data frames, Elementary statistics with R-Qualitative and Quantitative data, Numerical measures, probability distribution, interval estimation and simple linear regression.

08 Hours

List of Lab Experiments

Sl.No	Name of the Experiment
1	Introduction to R Software and basic commands
2	Demonstration and operations of Vectors
3	Operations of Matrices
4	Demonstration of Lists
5	Demonstration of Data Frames
6	Qualitative Data Analysis
7	Quantitative Data Analysis
8	Numerical Measures of Data
9	Probability Distribution
10	Linear Regressions

Course Outcomes:

On completion of this course, the students are able to :

- Find the probability using different distributions.
- Apply the stochastic process and Markov chain in prediction of future events.
- Calculate the various parameters of Queuing models.
- Analyze the data by using the concepts of sampling theory.
- Use the statistical software R for analyzing the data .

Text Books:

1. Dr. B.S. Grewal: "Higher Engineering Mathematics", Khanna publishers, New Delhi, 42nd Edition, 2012, ISBN: 9788174091956.

2. R.E.Walpole, R.H.Myers.R.S, L.Myers and K.Ye: "Probability and Statistics for Engineers and Scientists", Pearson Education, Delhi, 8th Edition, ISBN 13: 9780131877115.
3. W.N.Venables, D.M.Smith : "An introduction to R. R- manual".

Reference Books:

1. Erwin Kreyszig: "Advanced Engineering Mathematics", (Chapters 13,14,19,21,24,25), Wiley Pvt. Ltd., India, New Delhi, 9th Edition, 2011, ISBN 13: 9788126531356.
2. John Verzani: "Using R for introductory Statistics", Champan and Hall/ CRC, New York, Washington D.C., ISBN: 978-1-59327-384-2.
3. Sheldom M Ross: "Probability models for Computer Science", Academic Press, 2009, ISBN: 9780124079489.
4. Murray R Spiegel, John Schiler and Alu Srinivasan: "Probability and Statistics", Schaum's Outline series, 2nd Edition, ISBN: 9780071795579.

E-Resources:

1. http://www.zums.ac.ir/ebooks/mathematics/essential-engineering-mathematic.
2. https://archive.org/details/AdvancedEngineeringMathematics10thEdition
3. https://www.r-project.org/
4. www.r-tutor.com

Formal Languages and Automata Theory

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST42	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the basic concepts of finite automata.
- Learn the regular expressions, languages and their applications.
- Describe the context free grammars, languages and their applications.
- Acquire knowledge about push down automata, languages and its types.
- Introduce Turing machines.

Syllabus

Module - I

Introduction to Finite Automata: Introduction to Finite Automata, The central concepts of Automata theory, Deterministic finite automata, Nondeterministic finite automata. **08 Hours**

Module - II

Finite Automata, Regular Expressions: An application of finite automata, Finite automata with Epsilon-transitions, Regular expressions Finite Automata and Regular Expressions, Applications of Regular Expressions. **08 Hours**

Module - III

Regular Languages, Properties of Regular Languages: Regular languages, Proving languages not to be regular languages, Closure properties of regular languages, Decision properties of regular languages, Equivalence and minimization of automata. **08 Hours**

Module - IV

Context-Free Grammars and Languages: Context free grammars, Parse trees, Applications, Ambiguity in grammars and Languages.

Properties of Context-Free Languages: Normal forms for CFG's, Closure properties of CFL's. **08 Hours**

Module - V

Pushdown Automata: Definition of the Pushdown automata, The languages of a PDA, Equivalence of PDA's and CFG's, Deterministic Pushdown Automata.

Introduction to Turing Machine: The Turing machine. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Design the different types of automata for given regular expression and regular languages.
- Write the grammar for the given regular expressions.
- Draw the parse tree for the given context free grammars.
- Design the PDA for the given languages.
- Convert the given PDA to its equivalent CFG and vice versa.

Text Book:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: "Introduction to Automata Theory, Languages and Computation", (Chapters 1-8), Pearson New International Edition, 3rd Edition, 2014, ISBN-13: 978-0321455369.

Reference Books:

1. Peter Linz: "An Introduction to Formal Languages And Automata", Jones and Bartlett Publishers, 5th Revised Edition, 2011, ISBN-13: 978-1449615529, ISBN-10: 144961552X.
2. K.L.P. Mishra: "Theory of Computer Science, Automata, Languages, and Computation", PHI Learning, 3rd Edition, 2009, ISBN 10: 8120329686 ISBN 13: 9788120329683.

E-Resources:

1. http://www.ebook777.com/theory-finite-automata-introduction-format-languages.
2. http://www.techmela.ucoz.com/_id/o/22-introuctionto.pdf

Design and Analysis of Algorithms

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST43	3:0:0:0	3	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Acquire the knowledge of Algorithm and problem solving technique.
- Learn how to analyze the complexity of an algorithm in terms of time and space.
- Understand different techniques like divide and conquer, decrease and conquer etc., to solve problems.

Syllabus

Module - I

Introduction: What is an algorithm? Fundamentals of algorithmic problem solving, Fundamentals of the analysis of algorithm efficiency, Asymptotic Notations and basic efficiency classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms Brute Force Approaches: Introduction, Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching. **08 Hours**

Module - II

Divide and conquer: Divide and Conquer: General Method, Binary Search, Merge Sort, Quick Sort and its performance.

The greedy method: The General Method, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Single Source Shortest Paths. **08 Hours**

Module - III

Decrease and conquer approaches: Introduction, Insertion Sort, Depth First Search and Breadth First Search, Topological Sorting.

Transfer and conquer: Introduction, Balanced search trees, Heap and Heap sort.

08 Hours

Module - IV

Space-Time Trade-offs: Introduction, Sorting by Counting, Input Enhancement in String Matching (Horspool algorithm).

Dynamic programming: The General Method, Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem, The Travelling Salesperson problem, Computing a Binomial co-efficient. **08 Hours**

Module - V

Limitations of algorithmic power and coping with them: Lower-Bound Arguments, Decision Trees.

Backtracking: n - Queens problem, Subset – Sum Problem.

Hashing: Introduction, Open hashing, Closed hashing.

Branch and bound: Assignment problem, Knapsack problem. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Identify asymptotic notations and basic efficiency classes.
- Solve problems using various techniques like greedy and divide-and-conquer.
- Compute problems using various techniques like decrease-and-conquer and transfer-and-conquer.
- Use different algorithms like TSP, Floyd's etc. to solve real world problems.
- Develop solutions for n - Queens problem, Subset – Sum Problem, Assignment problem, Knapsack problem etc.

Text Books:

1. Anany Levitin: "Introduction to The Design and Analysis of Algorithms", (Chapters 1-5,7,9,11), Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", (Chapters 1,3-8,10-12), Universities Press, Hyderabad, 2nd Edition, 2007, ISBN: 10: 8173716129.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: "Introduction to Algorithms", PHI, London, England, 3rd Edition, 2010, ISBN: 9780262033848.
2. R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: "Introduction to the Design and Analysis of Algorithms A Strategic Approach", McGraw-Hill Higher Education, USA, International Edition, 2005, ISBN-13: 978-0071243469.

E-Resources:

1. http://www.pearsonhighered.com
2. http://www.ebooks.com
3. http://www.worldcat.org
4. http://www.citc.ui.ac.ir/zemoni/cls.pdf

Microprocessors (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI44	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course will enable students to :

- Understand the basics of 8086 microprocessor family and its architecture.
- Become aware of constructing the machine codes for 8086 instructions.
- Learn 8086 instruction sets as needed to solve programming problems.
- Acquire knowledge about signals, timing and system connections for a simple 8086-based microcomputer.
- Understand how the 8086 responds to interrupts, how interrupts-service procedures are written and operation of peripheral devices.

Syllabus

Module - I

Computers, Microcomputers and Microprocessors- An Introduction: Types of Computers, How Computers and Microcomputers are used, an example, Overview of Microcomputer Structure and Operation, Microprocessor Evolution and Types, The 8086 Microprocessor Family-Overview, 8086 Internal Architecture, Introduction to Programming the 8086. **08 Hours**

Module - II

8086 Family Assembler Language Programming-I: Constructing the machine codes for 8086 Instructions (MOV Instructions only), Assembly Language Program Development Tools, Data Transfer Instructions, Arithmetic Instructions. **08 Hours**

Module - III

8086 Family Assembler Language Programming-II: Bit-manipulation Instructions, String Instructions, Program-Execution Transfer Instructions, Processor Control Instructions. **08 Hours**

Module - IV

8086 System Connection Timing, Interrupts and Interrupt Applications: A Basic 8086 Microcomputer System, 8086 Bus activities during a read machine cycle, 8086 Bus activities during a write machine cycle, A closer look at the 8086, 8086 Interrupts and Interrupt Responses, Hardware Interrupt Applications, 8259 A Priority Interrupt Controller, Overview and System Connections, Software Interrupt Applications.

08 Hours

Module - V

Digital Interfacing: Programmable Parallel Ports and Handshake Input/Output, Methods of Data Transfer, Implementing Handshake Data Transfer, 8255A Internal Block Diagram and System Connections, 8255A Operational Modes and Initialization, Constructing and Sending 8255A Control Words. **08 Hours**

Laboratory

1. a) Search a key element in a list of 'n' 16-bit numbers using the Binary search algorithm.
   b) Read the status of eight input bits from the Logic Controller Interface and display 'FF' if the parity of the input read is even; otherwise display '00'.
2. a) Write two ALP modules stored in two different files; one module is to read a character from the keyboard and the other one is to display a character. Use the above two modules to read a string of characters from the keyboard terminated by the carriage return and print the string on the display in the next line.
   b) Implement a BCD Up-Down Counter on the Logic Controller Interface.
3. a) Sort a given set of 'n' numbers in ascending order using the Bubble Sort algorithm.
   b) Read the status of two 8-bit inputs (X and Y) from the Logic Controller Interface and display X*Y.
4. a) Display the message "Welcome To Microprocessor Lab" at the centre of the screen, clear the screen before displaying the message.
   b) 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).
5. a) Reverse a given string and check whether it is a palindrome or not.
   b) Assume any suitable message of 12 characters length and display it in the rolling fashion 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).
6. a) Read your name from the keyboard and display it at a specified location on the screen after the message "What is your name?", you must clear the entire screen before display.

b) Drive a Stepper Motor interface to 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).

7. a) Compute ${}^{n}C_{r}$ using recursive procedure. Assume that 'n' and 'r' are non negative integers.

   b) Drive a stepper motor interface to rotate by N steps clockwise direction and N steps counter-clockwise direction. (N is specified by the examiner) Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).

8. a) Generate the first 'n' Fibonacci numbers.

   b) Generate a Sine wave form using the DAC interface. (The output of the DAC is to be displayed on the CRO).

9. a) Read the current time from the system and display it in the standard format on the screen.

   b) Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).

10. a) Write a program to create a file (input file) and to delete an existing file.

    b) Generate the Fully Rectified Sine waveform using DAC interface (The output of the DAC is to be displayed on the CRO).

Course Outcomes:

On completion of this course, the students are able to :

- Discuss the microprocessor evolution, types and 8086 internal architecture.
- Construct the machine codes for 8086 instructions with the help of instruction templates.
- Describe functions of different types of 8086 assembly language instructions and use assembly language program to solve simple problems.
- Explain 8086 interrupt types and their applications.
- Describe the operations of peripheral devices.

Text Book:

1. Douglas V. Hall: "Microprocessors and Interfacing", (Chapter 2,3,5-9), TMH, New Delhi, Revised 2nd Edition, 2006, ISBN-13: 978-0070257429.

Reference Book:

1. Barry B. Brey: "The Intel Microprocessors", Pearson Education, New Delhi, 8th Edition, 2009, ISBN-13: 978-8131726228.

E-Resources:

1. http://www.nptel.ac.in/downloads/106108100/
2. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22
3. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22#tbm=bksandq=Bary+b+brey
4. http://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/Microprocessors%20and%20Microcontrollers/pdf/Teacher_Slides/mod1/M1L3.pdf
5. http://www.nptel.ac.in/downloads/106108100/
6. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22
7. https://www.google.co.in/search?tbo=pandtbm=bksandq=inauthor:%22Douglas+V.+Hall%22#tbm=bksandq=Bary+b+brey
8. http://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/Microprocessors%20and%20Microcontrollers/pdf/Teacher_Slides/mod1/M1L3.pdf

UNIX and Shell Programming (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI451	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course will enable students to **:**

- Understand the history, origin, features and architecture of UNIX Operating System.
- Learn basic commands to interact with UNIX System and VI editor.
- Understand UNIX file system.
- Learn shell scripting and Perl scripting.

Syllabus

Module - I

UNIX architecture: The operating system, the UNIX operating system, Linux and GNU. The UNIX architecture, features of UNIX, POSIX and Single UNIX specification, Internal and External commands, Command structure, man browsing and manual pages on-line.

The file system: The parent – child relationship, the HOME variable, pwd, cd, mkdir, absolute pathname, relative pathname.

vi editor: vi basics, input mod, saving text and quitting, searching for a pattern(| and ?), substitution- search and replace(:s). **08 Hours**

Module - II

Basic file attributes: ls: listing directory contents, the UNIX file system, ls –l, -d option, file ownership, file permissions, chmod, directory permissions, changing file ownership.

More file attributes: File systems and inodes, hard links, symbolic links and ln, the directory, umask, modification and access times, find.

The Process: Process basics, ps: process status, system processes(-e or –a), mechanism of process creation, process states and zombies, running jobs in background, nice: job execution, job control, at and batch, cron, time. **08 Hours**

Module - III

Simple filters: pr, head, tail, cut, paste, sort, uniq, tr.

Filters using regular expressions – grep and sed: grep, Basic Regular Expressions (BRE), Extended Regular Expressions (ERE) and egrep, sed: the stream editor, line

addressing using multiple instructions (-E and -F) context addressing, writing selected lines to a file (w), text editing, substitution (s), basic regular expression revisited.

08 Hours

Module - IV

The shell: The shell's interpretive cycle, shell offerings, pattern matching, escaping and quoting, redirection, pipes, tee, command substitution, shell variables.

Essential shell programming: Shell scripts, read using command line arguments, exit and exit status of command, the logical operators && and ||, the if conditional, using test and {} to evaluate expression. The case conditional, expr, $0, while, for, debugging shell scripts with set –x. **08 Hours**

Module - V

Perl – The master manipulator: Perl preliminaries, the chop function, variables and operators, the string handling functions, specifying file names in command line, $_, current line number ($.) and the range (..), list and arrays, foreach, split, join, dec2bin. pl, grep, associative arrays, regular expressions and substitutions, file handling, file tests, subroutines.

Networking Tools: TCP/IP basics, the applications. **08 Hours**

Laboratory

1. Execute the following list of basic commands in UNix:
   (i) pwd (ii) mkdir (iii) cd (iv) who (v) echo
   (vi) cat (vii) rm (viii) mv (ix) wc (x) cp
2. Execute the basic file attributes with all possible options:
   (i) ls (ii) chmod
3. Execute basic commands using vi editor:
   a. input mode commands
   b. saving text and quitting
   c. navigation
   d. editing text
   e. searching pattern
4. Execute the following filters using regular expressions with all possible options:
   (i) grep (ii) sed
5. Write a shell script to display current date and calendar.
6. Write a shell script to search for a pattern using grep in an employee database using interactive way.

7. Write a shell script to search for a pattern in an employee database using command line argument.
8. Write a shell script using test, $0 and $# in an if-elif-if construct.
9. Write a shell script using case, to offer 5 item menu:
   a. List of files
   b. Process of user
   c. Today's date
   d. Users of system
   e. Quit to UNIX
10. Write a shell script for matching pattern in student list using for loop.

Course Outcomes:

On completion of this course, the students are able to :

- Describe history, origin, feature and architecture of UNIX operating system.
- Interact with UNIX system easily.
- Construct and edit files, search for any patterns using regular expressions.
- Solve complex jobs using tools and utilities available in UNIX.
- Design and develop various tasks by using Shell and Perl scripting.

Text Book:

1. Sumitabha Das: "UNIX – Concepts and Applications", (Chapters 1,2,4,6-9,11-14,17,19), Tata McGraw Hill, Noida, 4th Edition, 15th Reprint, 2011, ISBN-13: 978-0-07-063546-3.

Reference Books:

1. Behrouz A. Forouzan and Richard F. Gilberg: "UNIX and Shell programming", Cengage Learning, India, 1st Edition, 2005, ISBN: 81-35-0325-9.
2. M G Venkatesh Murthy: "UNIX and Shell programming", Pearson Education, Delhi, 1st Edition, 2005, ISBN: 81-7758-745-5.

E-Resources:

1. http://www.mhhe.com/das/uca
2. http://www.tutorialspoint.com/unix/unix_tutorials.pdf.
3. http://www.perldoc.perl.org/

Object Oriented Programming with C++ (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI452	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FE

Course Objectives

This course will enable students to :

- Introduced to the basic principles of object oriented programming.
- Understand the features of C++.
- Learn the concepts of operator overloading and friend functions.
- Acquire the knowledge of inheritance to build C++ programming.
- Understand different techniques like exception handling, virtual functions etc.

Syllabus

Module - I

Introduction: Overview of C++, What is object oriented programming, Some C++ fundamentals, Overview of pointers and user defined types, Functions: argument passing, in line functions, function overloading.

Class and objects: Class Specification, Class Objects, Scope resolution operator, Access members, Defining member functions. **08 Hours**

Module - II

Classes and Objects-1: Constructors, Destructors, Parameterized constructors, Static data members and functions, Friend functions, Passing objects as arguments, Returning objects, Arrays of objects, Dynamic objects, Pointers to objects, Copy constructors, Generic functions and classes with examples. **08 Hours**

Module - III

Classes and Objects-2: Operator overloading using member functions such as +, - , pre-increment, post-increment, Operator overloading using friend functions such as +, - , pre-increment, post-increment [ ], <<, >>.

Inheritance: Base Class access control, Inheritance and protected members, Protected base class inheritance, Inheriting multiple base classes. **08 Hours**

Module - IV

Inheritance: Constructors, Destructors and Inheritance, Passing parameters to base class constructors, Granting access, Virtual base classes.

Virtual functions, Polymorphism: Virtual function, Calling a Virtual function through

a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, Pure virtual functions, Early v/s late binding. **08 Hours**

Module - V

I/O System Basics, File I/O: C++ stream classes, Formatted I/O, I/O manipulators, fstream and the File classes, File operations

Exception Handling: Exception handling fundamentals, Exception handling option, Overview of STL. **08 Hours**

Laboratory

List of Experiments:

1. Design, develop, and execute a program in C++ based on the following requirements: An EMPLOYEE class is to contain the following data members and member functions: Data members: Employee_Number (an integer), Employee_ Name (a string of characters), Basic_Salary (an integer) ,All_Allowances (an integer), IT (an integer), Net_Salary (an integer). Member functions: to read the data of an employee, to calculate Net_Salary and to print the values of all the data members. (All_Allowances = 123% of Basic; Income Tax (IT) = 30% of the gross salary (= basic_Salary _ All_Allowance); Net_Salary = Basic_Salary + All_ Allowances – IT).

2. a. Define a student class with USN, Name and Marks in 3 tests of a subject declare an array of 10 student objects using appropriate functions, Find the average of 2 better marks for each student, print the USN, Name and the average marks of all the students.

   b. Simulation of the above program using virtual lab.

3. Write a C++ program to create a class called complex and implement the following overloading function. ADD that return a complex number.

   a. ADD(a, S2)—where a is an integer(real part) and S2 ia a complex number.

   b. ADD(S1, S2)—where S1 and S2 are complex numbers.

4. Design, develop, and execute a program in C++ to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.

   i. STRING s1 = “NCET”

   ii. STRING s2 = “BENGALURU”

   iii. STIRNG s3 = s1 + s2; (Use copy constructor)

5. Design, develop, and execute a program in C++ to create a class called OCTAL, which has the characteristics of an octal number. Implement the following operations by writing an appropriate constructor and an overloaded operator +.

   OCTAL h = x ; where x is an integer

   int y = h + k ; where h is an OCTAL object and k is an integer.

   Display the OCTAL result by overloading the operator <<. Also display the values of h and y.

6. a. Write a C++ program to create a template function for quick sort and demonstrate sorting of integers and doubles.

   b. Simulation of the above program by using virtual lab.

7. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance create the classes UGSTUDENT and PGSTUDENT having fields as Semester, fees and stipend. Enter the data at least for 5 age for all UG and PG students separately.

8. Write a C++ program to create a class called MATRIX using a two dimensional array of integers. Implement the following operations by overloading the operator == which checks the compatibility of two matrices m1 and m2 to be added and subtracted. Perform the addition and subtraction by overloading the operators + and – respectively. Display the results by overloading operator <<.

```
if(m1==m2)
{
   m3=m1+m2;
   m4=m1-m2;
}
else
   Display error.
```

Course Outcomes:

On completion of this course, the students are able to :

- Describe the basic principles of object oriented programming language using C++.
- Illustrate class and objects, inheritance and polymorphism features, abstract classes.
- Construct C++ programs using operator overloading.

- Design and develop object oriented programming using inheritance.
- Interpret the basic concepts of I/O systems, file I/O and exception handling.

Text Book:

1. Herbert Schildt: "The Complete Reference C++", (Chapters 1-5), Tata McGraw Hill, US, 5th Edition, 2014, ISBN-10: 00716348 00, ISBN – 13: 9780071634809.

Reference Books:

1. Stanley B. Lippmann, Josee Lajore: "C++ Primer", Pearson Education, US, 5th Edition, 2012, ISBN 10: 8131775275, ISBN - 13: 9788131775271.
2. Paul J Deitel, Harvey M Deitel: "C++ for Programmers", Pearson Education, US, 7th Edition, 2010, ISBN 10: 0132165414, ISBN 13: 9780132165419.
3. K R Venugopal, Rajkumar Buyya, T Ravi Shankar: "Mastering C++", Tata McGraw Hill, New Delhi, 4th Edition, 1999, ISBN-10/AS1: 0074634542, ISBN-13: 9780074634547.

E-Resources:

1. https://docs.google.com/file/d/OByYLrayxuOPHyvJHCEFtcFFCNVE/edit.
2. http://books.google.co.in/books?
3. www.deitel@deitel.com
4. http://freedomputurbooks.com/masterin'sccp-by-venugopal.html
5. www.cplusplus.com
6. www.doc.ic.ac.uk
7. www.pearsonhighered.com/savitech

Introduction to Programming using Python (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI453	3:0:2:0	4	CIE:50 SEE:50	3 Hours	FE

Course Objectives:

This course enable students to :

- To get a clear understanding of Object Oriented Programming.
- To learn Python with a focus on regular expressions, exception handling, file handling, creating modules, interacting with database.

Syllabus

Module - I

Introduction and overview: What is Python, Origin, Comparison, Comments, Operators, Variables and Assignment, Numbers, Strings, Lists and Tuples, Dictionaries, if Statement, while Loop, for Loop and the range() built-in Function, Files and the open() Built-in Function, Errors and Exceptions, Functions, Classes, Modules.

Syntax and Style: Statements and Syntax, Variable Assignment, Identifiers, Basic Style Guidelines, Memory Management, Python Application Examples. **08 Hours**

Module - II

Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types. Numbers and Strings Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions.

Sequences: Strings, Lists, and Tuples, Sequences, Strings, Strings and Operators, String-only Operators, Built-in Functions, String Built-in Methods, Special Features of Strings. **08 Hours**

Module - III

Lists Operators, Built-in Functions, List Type Built-in Methods, Special Features of Lists, Tuples, Tuple Operators and Built-in Functions, Special Features of Tuples. Conditionals and Loops if statement, else if Statement, while Statement, for Statement, break Statement, continue Statement, pass Statement, else Statement.

08 Hours

Module - IV

Files and Input/output File Objects, File Built-in Function, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules.

Exception handling: The dir Function, Errors, Runtime Errors, The Exception Model, Exception Hierarchy, Handling Multiple Exceptions, raise, assert. **08 Hours**

Module - V

Regular Expressions: Introduction/Motivation, Special Symbols and Characters for REs, REs and Python.

Programming Exercise: Check for data error in CSV files: Numeric Check, Alphanumeric Check, Email Check, Date Check.

Database Interactions: Database Connection, creating database tables, insert data into table, reading, updating data. **08 Hours**

List of Experiments:

1. Create a new program called hello world.py. You will use this file to write your very first 'Hello world!' program.
2. Write a program using print that, when run, prints out a tic-tac-toe board.
3. Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3,1/4... 1/10.
4. Write a program using a for loop that calculates exponentials. Your program should ask the user for a base base and an exponent exp, and calculate baseexp.
5. Write a method fact that takes a number from the user and prints its factorial.
6. Write a function roots that computes the roots of a quadratic equation. Check for complex roots and print an error message saying that the roots are complex.

Course Outcomes:

On completion of this course, the students are able to :

- Describe basic of object oriented principles in Python.
- Explain type of operators and Built-in functions.
- Identify built-in libraries and conditional statements like loops.
- Interpret file handling and exception handling mechanisms.
- Use regular expressions and apply the same in solving specific problems.

Text Book:

1. Mark Lutz: "Learning Python", O'REILLY, 5th Edition, 2013, ISBN: 978-4493-5573-9.

Reference Books:

1. Barry, Paul: "Head First Python", O'REILLY, 2nd Edition, 2010, ISBN: 978-1-4493-8267-4.
2. David M. Beazley: "Python Essential Reference", Developer's Library, 4th Edition, 2010, ISBN: 0672329786.

E-Resources:

1. http://www.tutorialspoint.com/python/
2. https://www.codementor.io/learn-python-online
3. https://www.youtube.com/playlist?list=PL9FAE4422FA13FDE4

Introduction to Cyber Security and Cyber Laws

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST461	3:0:0:0	3	CIE:50 SEE:50	3 Hours	EE

Course Objectives:

This course enable students to :

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

Syllabus

Module - I

Introduction to Cyber crime: Cyber crime: Definition and Origins of the Word, Cyber crime and Information Security, Who are Cyber criminals?, Classifications of Cyber crimes, Cyber crime: The Legal Perspectives, Cyber crimes: An Indian Perspective, Cyber crime and the Indian ITA 2000, A Global Perspective on Cyber crimes, Cyber crime Era: Survival Mantra for the Netizens.

Cyber Offenses: How Criminals Plan Them: How Criminals Plan the Attacks, Social Engineering, Cyber stalking, Cyber cafe and Cyber crimes, Bot nets: The Fuel for Cyber crime, Attack Vector, Cloud Computing. **08 Hours**

Module - II

Cyber crime: 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/CellPhones, Mobile Devices: Security Implications for organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops. **08 Hours**

Module - III

Tools and Methods Used in Cyber crime: Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Key loggers and Spy wares, Virus and Worms, Trojan Horses and Back doors, 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

Module - IV

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, Anti forensics. **08 Hours**

Module - V

Cyber crimes and Cyber security: The Legal Perspectives, Introduction, Cyber crime and the Legal Landscape around the World, Why Do We Need Cyber laws: The Indian Context, The Indian IT Act, Challenges to Indian Law and Cyber crime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Amendments to the Indian IT Act, Cyber crime and Punishment, Cyber law, Technology and Students: Indian Scenario.

08 Hours

Course Outcomes:

On completion of this course, the students are able to :

- Acquire knowledge about the cyber security cyber crime and cyber offenses.
- Explain cyber crime on various mobile and wireless devices.
- Use of tools and methods in cyber crime and security.
- Interpret computer forensics.
- Understand legal issues in cyber crime.

Text Book:

1. Sunit Belapure and Nina Godbole: "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", (Chapter 1-7), Wiley India Pvt. Ltd., New Delhi, India, 2011, ISBN: 8126521791.

Reference Books:

1. Dr. Surya Prakash Tripathi, Ritendra Goyal, Praveen Kumar Shukla, KLSI: "Introduction to information security and cyber laws", Dreamtech Press, New Delhi, India, 2014, ISBN 13: 789351194736.

E-Resources:

1. http://www.civilserviceindia.com/subject/General-Studies/notes/basics-of-cyber security.html
2. http://uttaminstitute.ac.in/CYBER.pdf
3. http://www.vssut.ac.in/lecture_notes/lecture1423183198.pdf
4. http://www.tutorialspoint.com/information_security_cyber_law/introduction.html

Linear Integrated Circuits

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST462	3:0:0:0	3	CIE:50 SEE:50	3 Hours	EE

Course Objectives:

This course enable students to :

- Interpret the basic building blocks of linear integrated circuits.
- Learn the linear and non-linear applications of operational amplifiers.
- Acquire the knowledge about the theory and applications of analog multipliers and PLL.
- Understand the theory of ADC and DAC.
- Learn the concepts of waveform generation and introduce some special function ICs.

Syllabus

Module - I

IC Fabrication and Circuit Configuration for Linear IC: Advantages of ICs over discrete components, Manufacturing process of monolithic ICs , Construction of monolithic bipolar transistor, Monolithic diodes, Integrated resistors, Monolithic capacitors, Inductors, Current mirror and current sources, Current sources as active loads, Voltage sources, Voltage references, BJT Differential amplifier with active loads, General operational amplifier stages, and internal circuit diagrams IC 741, DC and AC performance characteristics, slew rate, Open and closed loop configurations.

08 Hours

Module - II

Applications of Operational Amplifiers: Sign changer, Scale changer, Phase shift circuits, Voltage follower, V-to-I and I-to-V converters, Adder, Subtractor, Instrumentation amplifier, Integrator, Differentiator, Logarithmic amplifier, Antilogarithmic amplifier, Comparators, Schmitt trigger, Precision rectifier, Peak detector, Clipper and Clamper, Low-pass, High-pass and band-pass butter worth filters.

08 Hours

Module - III

Analog Multiplier and PLL: Analog multiplier using emitter coupled transistor pair, Gilbert multiplier cell, Variable trans conductance technique, Analog multiplier ICs and their applications, Operation of the basic PLL, Closed loop analysis, Voltage

controlled oscillator, Monolithic PLL IC 565, Application of PLL for AM detection, FM detection, FSK modulation and demodulation and frequency synthesizing. **08 Hours**

Module - IV

Analog to Digital and Digital to Analog Converters: Analog and digital data conversions, D/A converter, specifications, weighted resistor type, R-2R ladder type, Voltage mode and current mode R – 2R ladder types, switches for D/A converters, high speed sample-and hold circuits, A/D converters, specifications, Flash type, Successive approximation type, Single slope type, Dual slope type, A/D converter using voltage-to-Time conversion, Over-sampling A/D converters. **08 Hours**

Module - V

Waveform Generators and Special Function ICs: Sine-wave generators, Multi vibrators and triangular wave generator, Saw-tooth wave generator, ICL8038 function generator, Timer IC 555, IC voltage regulators – Three terminal fixed and adjustable voltage regulators – IC 723 general purpose regulator – Monolithic switching regulator, Switched capacitor filter IC MF10, Frequency to voltage and voltage to frequency converters, Audio power amplifier, Video amplifier, Isolation amplifier, Opto-couplers and fibre optic IC. **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Implement linear and non-linear applications of OP-AMPs.
- Design applications using analog multiplier and PLL.
- Illustrate the working of ADC and DAC using OP-AMPs.
- Generate waveforms using OP-AMPs circuits.
- Analyze special function ICs.

Text Books:

1. Sergio Franco: “Design with operational amplifiers and analog integrated circuits”, (Chapters 1-4,6,7,9-11), Tata McGraw-Hill, 4th Edition, 2015, ISBN: 978-0078028168.
2. D. Roy Choudhry, Shail Jain: “Linear Integrated Circuits”, (Chapters 4-10), New Age International Pvt. Ltd., 4th Edition, 2010, ISBN 10: 8122430988 ISBN 13: 9788122430981.

Reference Books:

1. Gray and Meyer: " Analysis and Design of Analog Integrated Circuits", Wiley International, 5th Edition, 2014, ISBN 10 : 8126521481, ISBN 13: 978-81-265-2148-7.
2. Ramakant A. Gayakwad: "OP-AMP and Linear ICs", Prentice Hall / Pearson Education, 4th Edition, 2002, ISBN: 8120320581, 9788120320581.

E-Resources:

1. https://books.google.co.in/books?isbn=0070530440
2. https://books.google.co.in/books?isbn=8122414702

Control Systems

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CST463	3:0:0:0	3	CIE:50 SEE:50	3 Hours	EE

Course Objectives:

This course enable students to :

- Learn the concepts of block diagrams and transfer functions.
- Understand the characteristics of closed loop control systems including steady state and transient response.
- Acquire the knowledge about basic performance criteria for 1st and 2nd order systems.
- Understand the basic control system design methods including root-locus diagrams and frequency response methods.

Syllabus

Module - I

Modelling of Systems: Introduction to Control Systems, Types of Control Systems, Effect of Feedback Systems, Differential equation of Physical Systems -Mechanical systems, Friction, Translational systems (Mechanical accelerometer, systems excluded), Rotational systems, Gear trains, Electrical systems, Analogous systems.

08 Hours

Module - II

Block diagrams and signal flow graphs: Transfer functions, Block diagram algebra, Signal Flow graphs (State variable formulation excluded). **08 Hours**

Module - III

Time Response of feed back control systems: Standard test signals, Unit step response of First and second order systems, Time response specifications, Time response specifications of second order systems, Steady – state errors and error constants. Introduction to PID Controllers (excluding design). **08 Hours**

Module - IV

Stability analysis: Concepts of stability, Necessary conditions for Stability, Routh-stability criterion.

Root-Locus Techniques: Introduction, The root locus concepts. **08 Hours**

Module - V

Frequency domain analysis: Correlation between time and frequency response, Bode plots, Experimental determination of transfer functions, Assessment of relative stability using Bode Plots. Introduction to lead, Lag and lead-lag compensating networks (excluding design). **08 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Describe the engineering fundamentals about block diagram and transfer functions.
- Identify the characteristics of closed loop control systems.
- Examine time response of 1st and 2nd order feedback control systems.
- Estimate stability analysis using R-H criteria and root locus concepts.
- Construct correlation between time and frequency responses

Text Book:

1. J. Nagrath and M. Gopal: "Control Systems Engineering", (Chapter 2,5-8, 10), New Age International (P) Limited Publishers, Delhi, 5th Edition, 2008, ISBN: 978-184-8290037.

Reference Books:

1. K. Ogata: "Modern Control Engineering", Pearson Education Asia/ PHI, Delhi, 5th Edition, 2002, ISBN 10: 0-13-615673-8, ISBN 13: 978-0-13-615673-4.
2. Benjamin C. Kuo, John: "Automatic Control Systems", Wiley India Pvt. Ltd., 8th Edition, 2008, ISBN-10: 0471134767, ISBN-13: 978-047113476.
3. Joseph J Distefano III et al.,: "Feedback and Control System", P Schaum's Outlines, TMH, McGraw-Hill, New York, 2nd Edition, 2007, ISBN: 0-07-017052-5.

E-Resources:

1. http://research.iaun.ac.ir/pd/mahmoodian/pdfs/UploadFile_9809.pdf
2. http://www2.nuu.edu.tw/emo/e-teaching%20materials/Automatic%20Control.pdf
3. https://murdercube.com/files/Miscellaneous/Feedback%20and%20Control%20System.PDF

Design and Analysis of Algorithms Laboratory

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSL47	1:0:2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course enable students to :

- Understand and use asymptotic notations to analyze the performance of algorithms.
- Learn different sorting and searching techniques.
- Get exposed to various algorithm design techniques.

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

1. a. Write a C/C++ program to sort the elements by using quick sort method.
   b. Simulation of the above program by using virtual lab.
2. a. Write a C/C++ program to sort the elements by using merge sort method.
   b. Simulation of the above program by using virtual lab.
3. Obtain the Topological ordering of vertices in a given digraph.
4. a. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
   b. Perform Dijkstra single source shortest path on a graph using virtual lab.
5. Implement 0/1 Knapsack problem using Dynamic Programming.
6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.
7. Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.
8. a. Compute the transitive closure of a given directed graph using Warshall's algorithm.
   b. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.
9. a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
   b. Check whether a given graph is connected or not using DFS method.
10. Implement N Queen's problem using Back Tracking.

Course Outcomes:

On completion of this course, the students are able to :

- Solve problems by applying appropriate algorithms.
- Analyze the efficiency of various algorithms.
- Apply techniques of stacks and queues to solve problems.
- Develop a program that can be solved in many ways using different techniques.
- Identify and evaluate complex problems using principles of maths and engineering science.

Text Books:

1. Anany Levitin: "Introduction to the Design and Analysis of Algorithms", Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: "Fundamentals of Computer Algorithms", Universities Press, Hyderabad, 2nd Edition, 2007, ISBN-10: 8173716129.

E-Resources:

1. http://cs.gmu.edu/~pwiegand/cs483-Spring06/lecturenotes/cs483-l1pf.pdf
2. http://www.cs.cornell.edu/~kozen/papers/daa.pdf

Cloud Computing Foundations (IC)

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSI48	1:0:2:0	2	CIE:50 SEE:50	3 Hours	FC

Course Objectives:

This course enable students to :

- Explain the phases of transition from classic data center to virtual data center and then to the Cloud
- Explain the key characteristics, services, and deployment models of Cloud
- Describe the Cloud infrastructure components and service management processes
- Describe the Cloud security concerns and solutions
- List the key considerations for migration to the Cloud

Syllabus

Module - I

Journey to the Cloud: Drivers for cloud computing, cloud definition and characteristics, building cloud infrastructure – a phased approach from Classic data center to virtual data center to Cloud, virtualization and its benefits. **02 Hours**

Module - II

Cloud Computing Primer: Cloud computing characteristics, cloud deployment models private, public, hybrid and community cloud, cloud services – SaaS, PaaS, and IaaS, cloud economics and challenges. **03 Hours**

Module - III

Cloud infrastructure and Management: Cloud infrastructure framework and components, infrastructure management and service creation tools, cloud service management processes asset and configuration management, service catalog management, financial management, capacity, performance and availability management, incident, problem and compliance management. **04 Hours**

Module - IV

Cloud Security: Basic information security concepts, cloud security concerns and threats, security mechanisms in cloud at compute, storage, and network layer, Governance, Risk and compliance in Cloud. **02 Hours**

Module - V

Cloud Migration Considerations: Considerations for choosing right application and cloud model, service provider specific considerations, cloud adoption phases, Financial and technical feasibility assessment, migration and optimization considerations. **03 Hours**

Course Outcomes:

On completion of this course, the students are able to :

- Explain the phases of transition from classic data center to virtual data center and then to the Cloud.
- Explain the key characteristics, services, and deployment models of Cloud.
- Describe the Cloud infrastructure components and service management processes.
- Describe the Cloud security concerns and solutions.
- List the key considerations for migration to the Cloud.

Reference Books:

1. Thomas Erl: "Cloud Computing", Pearson Education, 1st Edition, 2014, ISBN-13: 978-9332535923.
2. Judith Hurwitz, Marcia Kaufman, Fern Halper: "Cloud Computing for dummies", Wiley, 1st Edition, 2009, ISBN-13: 978-0470484708.

E-Resources:

1. http://www.buyya.com/MasteringClouds/ToC-Preface-TMH.pdf

Soft Skills Development

Course Code	L:T:P:S	Credits	Exam marks	Exam Duration	Course Type
15CSH49	0:2:0:0	1	CIE:50 SEE:50	2 Hours	HSS

Course Objectives:

This course will enable students to :

- Improve the communication skills.
- Develop the art of presentation and writing effectively.
- Enhance the technical knowledge.

To improve the communication and presentation skills, every student has to give a seminar on technical topics assigned by the supervisors. Each course coordinator/ faculty members will be assigned with few students to guide and monitor the presentation. The presentation shall be for 15 minutes. A brief report on the seminar has to be submitted by the student to the concerned department after completion of the seminar. The report shall be signed by the supervisor and the Head of the concerned department.

The objective of the seminar is to introduce students to the major constituent of technology that is concerned with critically reading, understanding, summarizing, explaining and presenting existing technical topics. Students have to refer one or more topics that are assigned to them by their supervisors. The idea behind the seminar system is to familiarize student more extensively with the methodology of their chosen subject, allow them to develop presentation skills, and also interact with example of practical problems.

Course Outcomes:

On completion of this course, the students are able to :

- Get rid of stage fear and answer questions from audience.
- Communicate confidently and fluently.
- Comprehend and prepare reports effectively.