B. E. SYLLABUS

INFORMATION SCIENCE & ENGINEERING

III & IV SEMESTER

With
Scheme of Teaching
& Examination
<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Qualification</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dr. Balasubramani R</td>
<td>Ph.D</td>
<td>Professor &amp; HOD</td>
</tr>
<tr>
<td>2</td>
<td>Dr. Udaya Kumar K Shenoy</td>
<td>Ph.D</td>
<td>Professor</td>
</tr>
<tr>
<td>3</td>
<td>Karthik Pai B H</td>
<td>M.Tech</td>
<td>Asst. Prof Gd III</td>
</tr>
<tr>
<td>4</td>
<td>Ashwini B</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>5</td>
<td>Deepa</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>6</td>
<td>Devidas</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>7</td>
<td>Rashmi Naveen</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>8</td>
<td>Vasudev Pai</td>
<td>M.Tech</td>
<td>Asst. Prof Gd II</td>
</tr>
<tr>
<td>9</td>
<td>Pranesh</td>
<td>M.Tech</td>
<td>Asst Prof Gd-II</td>
</tr>
<tr>
<td>10</td>
<td>Rakesh Joshi U</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>11</td>
<td>Manasa</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>12</td>
<td>Savita Sthawarmath</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>13</td>
<td>Abhir Bhandary</td>
<td>M.Tech</td>
<td>Asst. Prof Gd I</td>
</tr>
<tr>
<td>14</td>
<td>Srikanth Bhat. K</td>
<td>M.Tech</td>
<td>Asst Prof Gd I</td>
</tr>
<tr>
<td>15</td>
<td>Jason Elroy Martis</td>
<td>M.Tech</td>
<td>Asst Prof Gd I</td>
</tr>
<tr>
<td>16</td>
<td>Sunil Kumar Aithal</td>
<td>M.Tech</td>
<td>Asst Prof Gd I</td>
</tr>
<tr>
<td>17</td>
<td>Shwetha Bhat M</td>
<td>B.E</td>
<td>Asst Prof Gd I</td>
</tr>
</tbody>
</table>
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING  
NMAMIT, Nitte

Vision:  
The vision of the Dept. of ISE is to uniquely position the Dept. as a leader in innovation and excellence in information science and engineering through education, research and scholarship in a professional framework by addressing evolving global needs. Also the Dept. aims at creating top quality successful and sustainable programs and curricula for the students to address the emerging educational challenges and market demands.

Mission:  
• To provide outstanding education and research training to the students for their productive careers in industry, academia and government.  
• To provide a learning environment that promotes excellence and innovation, ethical practice and responsibility towards society.  
• To prepare the students to practice their professions competently to meet the ever-changing needs of society and to continue learning their discipline, allowing them to move into other related fields.  
• To promote active learning, critical thinking, and engineering judgement coupled with business and entrepreneurial skills.

Program: B.E. Information Science & Engineering

Programme Educational Objectives (PEO’s)  
• Graduates must gain both theoretical and practical knowledge to identify, formulate & solve challenges in Information Science & Engineering problems.  
• Graduates must work productively as Information Science Engineers, including supportive and leadership roles on multidisciplinary teams.  
• Graduates must communicate effectively, recognize and incorporate societal needs and constraints in their professional
endeavors, and practice their profession with high regard to legal and ethical responsibilities.

- Graduates must engage in life-long learning, such as graduate study, to remain current in their profession and be leaders in our technological society.

**Programme Outcomes (PO’s)**

BE (ISE) Engineering Program students must attain the following outcomes at the end of the course.

a. An ability to apply knowledge of mathematics, science and engineering

b. An ability to design and conduct experiments, as well as to analyze and interpret data

c. An ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability

d. An ability to function on multidisciplinary teams

e. An ability to identify, formulate and solve engineering problems

f. An understanding of professional and ethical responsibility

g. An ability to communicate effectively

h. The broad education necessary to understand the impact of engineering solutions in global, economic environmental and societal context

i. A recognition of the need for and an ability to engage in lifelong learning

j. A knowledge of contemporary issues

k. An ability to use the techniques, skills and modern engineering tools necessary for engineering practice and

l. An ability to strengthen the knowledge and understanding in the areas of computer networking and software engineering.
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
SCHEME OF TEACHING

Credits for Theory – Lab – Tutorials
(1 Theory hr. = 1 Credit, 2 Lab hrs. = 1 credit, 2 Tutorial hrs. = 1 credit)

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>29 Hours/week</th>
<th>CIE</th>
<th>SEE</th>
<th>Total Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>13IS301</td>
<td>Generating Functions and Transform Techniques</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>13IS302</td>
<td>Logic Design and Electronic Circuits</td>
<td>3+2+2</td>
<td>50</td>
<td>50</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>13IS303</td>
<td>Discrete Mathematical Structures and Number Theory</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>13IS304</td>
<td>Computer Organization and architecture</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5.</td>
<td>13IS305</td>
<td>Data Structures</td>
<td>3+0+2</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6.</td>
<td>13IS306</td>
<td>Object Oriented Programming</td>
<td>3+0+2</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>29</td>
<td>300</td>
<td>300</td>
<td>25</td>
</tr>
</tbody>
</table>
# DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

## SCHEME OF TEACHING

### IV Semester

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Subject Code</th>
<th>Name of the Subject</th>
<th>Contact hours/week</th>
<th>CIE</th>
<th>SEE</th>
<th>Total Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>13IS401</td>
<td>Probability Theory and Numerical Methods</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>13IS402</td>
<td>Analysis and Design of Algorithms</td>
<td>3+0+2</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>13IS403</td>
<td>Theory of Computations</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>13IS404</td>
<td>Data Communication</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5.</td>
<td>13IS405</td>
<td>Advanced Data Structures</td>
<td>4+0+0</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6.</td>
<td>13IS406</td>
<td>Unix and Shell Programming</td>
<td>3+0+2</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>7.</td>
<td>13IS407</td>
<td>Soft Skills (IEL)</td>
<td>0+4+0</td>
<td>50</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>30</strong></td>
<td><strong>350</strong></td>
<td><strong>350</strong></td>
</tr>
</tbody>
</table>

Note: The subjects with combined theory and lab, students must score minimum passing marks in each of the component.
GENERATING FUNCTIONS AND
TRANSFORM TECHNIQUES

Subject Code : 13IS301          Credits : 04
Hrs/Week    : 4              Total Hours : 52

UNIT - I
Introduction to Graph Theory: Definitions and examples, Subgraphs, Complements, and Graph Isomorphism, Euler Trails and Circuits, Hamiltonian paths and Cycles, Planar Graphs, Graph Colouring, Trees: Definitions, Properties and Examples. 10 Hrs

UNIT - II
Fourier Analysis: Periodic functions, Euler’s formulae, Fourier series of odd and even functions, functions with arbitrary period, half range series. Harmonic Analysis. Fourier integral theorem, Fourier Transforms, Inverse Fourier transform, Convolution theorem and Parseval’s identity. Fouriersine and Fourier cosine transforms, Inverse Fourier sine and Inverse Fourier cosine transforms. 12 Hrs

UNIT - III
Z transforms: Z-transform, standard forms, linearity property, damping rule, shifting rule. Inverse Z-transform, Finite differences and difference equations, Solving Difference equations using Z-transforms. 10 Hrs

UNIT - IV
Combinations with Repetition, Non-negative integer solution for linear equation, positive integer solution for linear equation. Fibonacci numbers, Catalan Numbers, The Principle of Inclusion and Exclusion, Generalization of the Principle, Derangements – Nothing is in its Right place, Rook Polynomials. 10 Hrs

UNIT - V
Generating Functions: Definition and Examples – Calculational Techniques, Partitions of Integers, the Exponential Generating Function, and the Summation Operator. The method of Generating Functions to solve Recurrence relations. 10 Hrs

TEXT BOOKS:
3. Narsing Deo, Graph theory
4. Graham knuth & Patashnik, Concrete Mathematics

**REFERENCE BOOKS:**
2. Harary, Graph theory, Narosa Publishing House, 1988

**LOGIC DESIGN AND ELECTRONIC CIRCUITS**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Credits</th>
<th>Hrs/ Week</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS302</td>
<td>03+01</td>
<td>3+2</td>
<td>39</td>
</tr>
</tbody>
</table>

**UNIT - I**

**Diode- Application:** Clippers, Clampers.

**Transistors:** Characteristics. Biasing: Fixed bias, emitter bias & Voltage divider bias

**Operational Amplifiers:** Voltage follower, Inverting and Non-Inverting amplifier Integrator, differentiator, summer and Schmitt Trigger.

**Oscillators:** Condition for oscillations. RC- Phase shift Oscillator, Crystal Oscillator.

8 Hrs

**UNIT - II**


**Data-Processing Circuits:** Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD-to-Decimal Decoders, Seven segment Decoders, Encoders, EX-OR gates, Parity Generators and Checkers, Magnitude Comparator, Read-Only-Memory, Programmable Array Logic, Programmable Logic, HDL Implementation of Data Processing Circuits.
Arithmetic Circuits: Binary Addition, Binary Subtraction, Unsigned Binary Numbers, Sign-Magnitude Numbers, 2’s Complement Arithmetic, Arithmetic Building Blocks, The Adder-Subtractor, Fast Adder, Arithmetic Logic Unit, Binary Multiplication and Division, Arithmetic Circuits using HDL.

12 Hrs

UNIT - III
Clocks and Timing Circuits: Clock Waveforms, Pulse forming circuits, TTL clock. 555 timer Astable and Monostable.

7 Hrs

UNIT - IV
 Registers: Types of Registers, Serial In – Serial Out, Serial In – Parallel Out, Parallel In– Serial Out, Parallel In – Parallel Out, Applications of Shift Register, Register Implementation in HDL.
 Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus, Decade Counters, Presettable Counters, Counter Design as a Synthesis Problem, Counter Design Using HDL.

7 Hrs

UNIT - V
Design of Sequential Circuit: Design and Analysis of Synchronous and Asynchronous Sequential Circuits with examples.


5 Hrs

TEXT BOOKS:
1. Electronic principles, Albert Malvino David J, Bates

REFERENCE BOOKS:
LABORATORY COMPONENTS

Hrs/Week : 02
Total Hours : 26 (13 weeks)

Credit : 01

PART — A

1. Half wave and Full wave (bridge) rectifier with and without Capacitor filter. To study the waveforms and to calculate the Ripple Factor and the Efficiency.
2. Design of Op-Amp circuits to work as:
   i) Voltage follower.  ii) Inverting / Non – Inverting Amplifier.
   iii) Integrator.
4. Design and implement an Astable Multivibrator for a given frequency and duty cycle using 555 timer.
5. Design of +8V regulator using +5V, three terminal IC regulator and determination of Load regulation.

PART — B

1. Design and implementation of a Half-adder and a Full-adder using minimum number of 2-input NAND gates.
2. Given any four variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
3. Design and implementation of a Full adder and a Full subtractor using 3:8 decoder and 4 input NAND gates.
4. Design and implementation of a Mod-N (N<8) Synchronous up counter using J-K flip flop ICs.
5. Design and implementation of the following using 4-bit shift register
   i) Ring counter    ii) Johnson counter
6. Design and implementation of an Asynchronous counter using a Decade counter IC to count up from 0 to n (n<=9).
PART — C

Using PSpice simulate the following:
1. Simulation of Half wave and Full wave (bridge) rectifier with and without Capacitor filter.
2. Simulation of Op-Amp circuits to work as:
   i) Voltage follower.
   ii) Inverting / Non – Inverting Amplifier.
   iii) Integrator.

Using Verilog/VHDL, Simulate the following:
4. Simulation of 8:1 multiplexer.
5. Simulation of full adder.
6. Simulation of the following:
   a. Ring counter   b. Johnson counter

Note: Student has to implement 2 questions. (One from either PART-A or PART-B and the other from PART-C)

DISCRETE MATHEMATICAL STRUCTURES AND NUMBER THEORY

Subject Code : 13IS303  Credits : 04
Hrs/ Week : 4  Total Hours : 52

UNIT - I

Set Theory and Counting:
Sets and subsets, Operations on sets, Permutations, Combinations, The Pigeonhole principle and Recurrence Relations.

Relations and its Properties:
Product sets and Partitions, Relations and Digraphs, Paths in relations and Digraphs  10 Hrs
UNIT - II
Relations and its Properties Contd:
Properties of relations, Equivalence relations, Computer representation of Relations and Digraphs and Transitive closure and Warshall’s algorithm.

Functions:
Definition, Types of functions, Invertible functions, Functions for computer science and Permutation functions. 10 Hrs

UNIT - III
Order relations and Structures:
Partially Ordered Sets, External elements of Partially ordered sets, Lattices, Semigroups and Groups. Semigroups, Isomorphism and homomorphism, Groups-Abelian groups 10 Hrs

UNIT - IV
Fundamentals of logic:
Propositions and Logical Operations, Conditional Statements, Biconditional statements, The laws of logic, Logical implication-Rules of inference, Quantifiers, Mathematical induction. 10 Hrs

UNIT - V
Introduction to number Theory:
Prime and Relatively prime numbers, Properties of integers, Modular arithmetic, Fermat’s and Eulers Theorems, Testing for primality, Euclids Algorithm, The Chinese Remainder Theorem. 12 Hrs

TEXT BOOKS:

REFERENCE BOOKS:
# COMPUTER ORGANIZATION AND ARCHITECTURE

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Credits</th>
<th>Hrs/Week</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>13IS304</td>
<td>04</td>
<td>4</td>
<td>52</td>
</tr>
</tbody>
</table>

## UNIT - I

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

**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

10 Hrs

## UNIT - II


10 Hrs

## UNIT - III

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

10 Hrs

## UNIT - IV

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

12 Hrs

## UNIT - V

**Basic Processing Unit:** Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, and Micro programmed Control

10 Hrs

## TEXT BOOK:

REFERENCE BOOKS:

DATA STRUCTURES
Subject Code : 13IS305 Credits : 03
Hrs/Week : 3 Total Hours : 39

UNIT - I
ARRAYS, STRINGS, STRUCTURES AND UNIONS:
Review of arrays, Multidimensional arrays, Sparse matrices, strings and structures, structure and functions, union, bit fields.

POINTERs:
Introduction to pointers, Pointers and functions, Pointers and Arrays, Pointers and Strings, Pointers and Structures

DYNAMIC MEMORY ALLOCATION:
Definition, Dynamic memory allocation functions, Programming examples

UNIT - II
DATA STRUCTURE:
Definitions, concept, overview and implementation of Data Structures

STACK:
Definition and Examples, Primitive operation, Examples, Representing Stacks in C: Implementation of push, pop and display operations on stack, testing for exceptional conditions, Expressions: Infix, postfix, prefix, Basic definitions and examples, Applications of stack- Checking the validity of a mathematical expression, Evaluating postfix expression, converting an expression from infix to postfix, converting an expression from infix to prefix, reversing a string using stack, Validity of a comment statement in C using stack
UNIT - III

RECURSION:
Recursive Definition and Processes, Recursion in C, Writing Recursive Programs, Tower of Hanoi, Efficiency of Recursion

QUEUES:
Definition and its sequential implementation, Types of queues, Circular Queue, Implementation of Circular Queue- two methods, Priority Queue and its implementation, Double ended queue, Application of queues – Round robin algorithm 8 Hrs

UNIT - IV

LIST:
Definition, Linked implementation of Stacks and Queues, Header nodes, Array implementation of list, Array vs Dynamic implementation, Non integer and non homogeneous lists, Operations on list like searching, updating, insert / delete at a specific position, after a node, insert/ delete into an ordered list, sorting a list 7 Hrs

UNIT - V

LIST continued:
Doubly linked list and its operations, Circular list and its operations, application of lists - polynomial representation,

FILE MANAGEMENT:
Defining and opening file, closing file, I/O operations on files, Error handling during file operations, random access to files, command line arguments, case study. 7 Hrs

TEXT BOOKS:
2. Programming in ANSI C, E.Balaguruswamy Mcgraw Hill Publishing. (For Unit 1 and File Management Chapter in Unit 5 in syllabus)
3. Understanding Pointers in C, Yeshavanth Kanetkar, BPB publications. (For Unit 1 in syllabus)
4. Classic Data Structures, D. Samanta, Phi Learning (2009) (Chapter 1, 2.4, 3.6.2, 5.5.3)

REFERENCE BOOKS:
LABORATORY COMPONENTS

Hrs/Week : 2  
Total Hours : 26 (13 weeks)

The students need to write, execute and test programs covering the entire syllabus of Data Structures.
Typical programs would be:
1. Managing records using structures, pointers – writing functions for insert, search and other operations. Same program to be implemented using dynamic memory allocation.
2. Dynamic memory allocation programs – searching, sorting, sum of n numbers.
3. Stack – Basic operations, infix to postfix, infix to prefix, evaluating a postfix expression, validity of a mathematical expression.
4. Recursive program examples including Tower of Hanoi.
5. Queues – Ordinary Queue, circular queue, priority queue, double ended queue implementation.
6. Linked list creation, insertion /deletion and display using pointers.
7. Linked implementation of stacks and queues.
8. Doubly linked list and Circular List implementation.
9. Files – managing records using files, command line arguments. Programs need to be written in C programming language.

OBJECT ORIENTED PROGRAMMING

Subject Code : 13IS306  
IA Marks : 50
Hrs/Week : 03  
Exam Hours : 3
Total Hours: 39  
Exam Marks : 50

UNIT - I

Principles of object – oriented programming
A look at Procedure Oriented Programming, Object Oriented Programming Paradigm, Basic Concepts of OOP, Benefits of OOP, Object oriented languages, Application of OOP.

Beginning with c++
What is c++, Applications of C++, Structure of C++ program, Basic Data types, derived data types, user defined data types, variables in C++, dynamic initialization of variables, reference variables, operators in c++, scope resolution operator, memory management operators, type cast operators, manipulators, namespace.  

7 Hrs
UNIT - II

Functions in C++:
Function prototyping, Inline Functions, Default Arguments, Function Overloading.

Classes and objects:
Introduction, C Structure Revisited, Specifying a Class, Defining Member Functions, Static Data Members, and Static Member Functions. Arrays of Objects, Objects as Functions Arguments, this pointer, Friend Functions, Returning Objects, Constant Member Functions.

UNIT - III

Constructors and Destructors
Introduction, Constructors, Parameterized Constructors, Multiple Constructors in a Class. Constructors with Default Arguments, Copy Constructors, Dynamic Constructors, Constant Objects, Destructors.

Operator Overloading and type Conversions:
Introduction, Defining Operator Overloading, overloading the Various Operators, Overloading the Increment and the Decrement Operators (Prefix and Postfix), Overloading the Unary Minus and the Unary Plus Operator, Overloading the Arithmetic Operators. Overloading the Relational Operators, Overloading the Assignment Operator, Overloading the Insertion and Extraction Operators, Rules for overloading operators. Type conversions.

UNIT - IV

Inheritance:
Introduction, Defining Derived Classes, single Inheritance, Protected Access Specifier, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Hybrid Inheritance, Virtual Base Classes, Abstract Classes, Constructors in Derived Classes, Nesting of Classes.

Pointers, Virtual Functions and Polymorphism
Introduction, Pointers, Pointers to objects, pointers to derived classes, Virtual Functions, Pure Virtual Functions.

UNIT - V

Templates
Introduction, Template Specifications, typename, Name Resolution Function Templates, Class Templates, Overloading of template Function, Advantages of Templates.

Exception Handling
Basics of exception Handling, Exception Handling Mechanism, Limitation of Exception Handling.
TEXT BOOK:

REFERENCE BOOKS:
1. Robert Lapore: Object-Oriented Programming in turbo C++
3. K.R. Venugopal: Mastering

LABORATORY COMPONENTS

Hrs/Week : 2
Total Hours : 26 (13 weeks)

Students have to write, execute and test programs covering the syllabus of IS306. Typical programs:
1. Program highlighting inheritance feature
2. Program using virtual functions
3. Program highlighting polymorphism feature
4. Program for string handling
5. Program for file handling
6. Programs using Functions & Operator Overloading

PROBABILITY THEORY AND NUMERICAL METHODS

Subject Code : 13IS401
Hrs/Week : 4
Total Hours : 52

UNIT - 1
Introduction to probability: finite sample space, conditional probability and independence. Baye’s theorem. One dimensional random variable, pdf, cdf, expectation and variance. Two and higher dimensional random variables, joint pdf, marginal pdf, covariance, correlation coefficient. 12 Hrs
UNIT – II
Probability Distributions and curve fitting: Binomial, Poisson, uniform, normal, Chi-square and exponential. Simple problems. Linear and quadratic curve fitting using least square principle. correlation and regression. 10 Hrs

UNIT – III
Sampling theory-Random samples, sampling distributions, t & F distributions. Moment generating function, Central limit theorem and its applications. 10 Hrs

UNIT – IV
Numerical Analysis: Finite differences, Newton-Gregory forward and backward difference interpolation formulae, Lagrange’s interpolation formula, Lagrange’s Inverse interpolation formula. Numerical differentiation using Newton’s forward & backward formulae. Numerical integration: General quadrature formula, Trapezoidal rule, Simpson’s one third rule, Simpson’s three eighth rule. 12 Hrs

UNIT – V
Roots of algebraic and transcendental equations, Numerical solution of first order ordinary differential equations: Taylor’s series Method, Modified Euler’s method, Runge–Kutta 4th order Method. Milne’s predictor-corrector methods. 8 Hrs

TEXT BOOKS:

REFERENCE BOOKS:
ANALYSIS AND DESIGN OF ALGORITHMS

Subject Code : 13IS402  Credits : 03
Hrs/Week : 3  Total Hours : 39

UNIT - I


UNIT - II

Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search.

Divide and Conquer: Merge sort, Quick sort, Binary Search, Binary tree traversals and related properties, Multiplication of large integers and Strassen’s Matrix Multiplication.  7 Hrs

UNIT - III


Transform and Conquer: Presorting, Balanced Search Trees, Heaps and Heap sort, Problem Reduction.  8 Hrs

UNIT - IV


Dynamic Programming: Computing a Binomial Coefficient, Warshall’s and Floyd’s Algorithms, The Knapsack Problem and memory functions.  8 Hrs

UNIT - V

Greedy Technique: Prim’s Algorithm, Kruskal’s Algorithm, Dijkstra’s Algorithm, Huffman Trees.

Backtracking: n queens problem, subset-sum problem.

Branch and Bound: Assignment problem, Knapsack problem. Definition of NP Completeness.  8 Hrs
TEXT BOOK:

REFERENCE BOOKS:

LABORATORY COMPONENTS

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>02</th>
<th>Credit</th>
<th>01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hours</td>
<td>26 (13 weeks)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Students have to write, execute and test programs covering the syllabus of IS405. The execution time of programs, for various types of inputs, have to be recorded and analyzed. Various versions of programs to solve same problem must be attempted and a comparison of performance must be drawn.

Typical problems that may be tried are
1. Merge sort
2. Quick sort
3. Topological sort in graphs
4. BFS, DFS algorithm and its applications
5. Heap sort
6. Prim’s algorithm
7. Kruskal’s algorithm
8. Dijkstra’s algorithm
9. Warshal’s algorithm
10. Floyd algorithm
11. N-Queens problem

Students may implement the programs using C/C++ language on Windows platform.
THEORY OF COMPUTATIONS

Subject code : 13IS403
Hrs/Week : 04
Total Hours : 52

LTP : 4-0-0
CIE : 50
SEE : 50

UNIT - I

Introduction to the theory of computation:
Mathematical preliminaries and notation, Three basic concepts
Some applications. 3 Hrs

Finite Automata:
Deterministic Finite accepter-Deterministic accepter and transition
graphs. Languages and DFA’s, Regular languages.
Nondeterministic finite accepter: Definition, Examples.
Equivalence of Deterministic and Nondeterministic Finite
Accepter. Reduction of the number of states in finite automata.
7 Hrs

UNIT - II

Regular Languages and Regular Grammars:
Regular expressions, Languages associated with regular
expressions, Regular expressions denote regular languages,
Regular expressions for regular languages, Regular grammars-
Right and Left linear grammars. Right and Left Linear grammar
examples. Right linear grammar generates regular languages, Right
linear grammars for regular languages. 7 Hrs

Properties of Regular Languages:
Closure properties of regular languages-Closure under simple set
operations. Closure under other operations. Identifying non regular
languages-Using the A pumping lemma. 4 Hrs.

UNIT - III

Context-Free languages.
Context free grammars – examples, Leftmost and Rightmost
derivations. Derivation Trees. Parsing and ambiguity-Ambiguity in
grammars and languages. 3 Hrs
**Simplification of CFG and Normal Forms:**
Methods for transforming grammars-Substitution rule. Removing useless, lambda, unit productions. Normal forms-Chomsky normal form and Greibach normal form. **7 Hrs**

**UNIT - IV**

**Properties of Context-Free Languages:**
Two pumping lemmas: A pumping lemma for Context-Free languages. A pumping lemma for Linear languages. **4 Hrs**

**Pushdown Automata**
Nondeterministic pushdown automata-Definition, Language accepted by a PDA. PDA for Context Free languages, Context free grammars for pushdown automata, Deterministic pushdown automata. **6 Hrs**

**UNIT - V**

**Turing Machines:**
The standard Turing machine-Definition, examples, Turing machine as language accepter. Turing machine as Transducers. Combining Turing machines for complicated tasks. Universal turing machine. Other models of Turing machines: Multitape turing machines, Nondeterministic turing machines etc. **6 Hrs**

**A Hierarchy of Formal Languages and Automata:**
Recursive and recursively enumerable languages, The Chomsky hierarchy.

**Limits of Algorithmic Computation:**
Some problems that cannot be solved by Turing machines, The post correspondence problem. **5 Hrs.**

**Text Books:**

**Reference Books:**
DATA COMMUNICATION

Subject Code : 13IS404  
Credits : 04
Hrs/Week : 4  
Total Hours : 52

Course prerequisites: Nil

UNIT - I

Introduction to data communications:
Communication Networks and Services: Telegraph Networks and Message Switching, Telephone Networks and Circuit Switching, the Internet, Computer Networks and Packet Switching.

Applications and Layered Architectures:
10 Hrs

UNIT - II

Digital Transmission Fundamentals – I:
10 Hrs

UNIT - III

Digital Transmission Fundamentals – II:
11 Hrs

UNIT - IV

Multiplexing and Peer to Peer Protocols:
Multiplexing: Frequency Division Multiplexing, Wavelength Division Multiplexing, Time Division Multiplexing. SONET: SONET Multiplexing, SONET frame structure. Framing, HDLC Data Control,

UNIT - V

Medium Access Control:

11 Hrs

TEXT BOOKS:

REFERENCE BOOKS:

ADVANCED DATA STRUCTURES

Subject Code : 13IS405
Hrs/Week : 4
Total Hours : 52

UNIT - I

Binary Trees: Operation on binary trees, application of binary trees.
Binary Tree representation: Node representation of Binary tree, Internal and external nodes, Implicit array representation of Binary trees, Binary tree traversals in C, non recursive in order tree traversal, algorithms on binary tree operations, Binary Search tree and Operations on Binary search tree, Inserting into Binary search tree, Deletion of a node in a Binary search tree.  

9 Hrs
UNIT - II
Trees: Threaded binary tree and its implementation, Traversal using Father Field, Heterogeneous Binary trees.
Example: The Huffman Algorithm.
Representing Lists as Binary Trees: Finding the Kth element, Deleting an element, Implementing Tree-Represented List, Constructing a Tree Represented List.
Trees and their application: C representation of trees, converting tree or forest into binary tree, tree traversals, General expression as a Tree, Evaluating an expression Tree, Constructing a Tree. 10 Hrs

UNIT - III
Searching: Sequential Searching, searching an ordered table, Indexed Sequential search, Interpolation search.
Sorting: Selection Sort, Binary Tree Sort, Insertion Sort, Shell Sort, Address calculation sort, Radix Sort.
Hashing: Definition, Hash Tables, Hash Functions, Hash collision, Resolving collision, Open Address Hashing, Chaining – coalesced Hashing and separate chaining. 11 Hrs

UNIT - IV
General search trees: Multiway search trees, B trees and B+ trees.
Balanced Trees: Red Black Tree, Splay Tree.
Other Trees: Binary Heaps, its operations, Leftist Heap, Skew Heap, Binomial Trees & Binomial Heap, Fibonacci Heap. 11 Hrs

UNIT - V
Fundamental file structure concepts, managing files of records
Field and record organization file access and file organization.
Organization of files for performance and indexing
Data compression, Reclaiming space in files, internal sorting and binary searching, key sorting; What is an index? A simple index for entry-sequenced file. 11 Hrs

TEXT BOOKS:
REFERENCE BOOKS:

UNIX AND SHELL PROGRAMMING
Subject Code : 13IS406       Credits : 03
Hrs/Week      : 3            Total Hours : 39

UNIT - I
Background and some basic commands

cal: The Calendar, date: Displaying and System date, echo: Displaying a Message, printf: An Alternative to echo, bc: The Calculator, script: Recording a session, passwd: Changing your password, who: who are the users?, uname: Knowing your machine’s characteristics, tty: Knowing your terminal, stty: Displaying and setting Terminal characteristics.

The filesystem and some file handling commands

cat: Displaying and creating Files, cp: Copying a File, rm: Deleting files, mv: Renaming files, more: Paging output, The lp Subsystem: Printing a file, file: Knowing the file types, wc: Counting lines, words and characters, od: Displaying data in octal, cmp: Comparing two Files, comm.: What is common?, diff: Converting one file to other, dos2unix and unix2dos: Converting between DOS and UNIX, Compressing files, gzip, gunzip, zip and unzip commands.
The **vi Editor**

Vi Basics, Input mode- Entering and replacing text, Saving text and Quotimg- The ex mode, Navigation, Editing Text, Undoing last editing instructions (u and U), Repeating the last command(.), Searching for a pattern(/ and ?), Substitution- Search and Replace(:s), Customizing vi.

**THE SHELL**

The shell’s interpretive cycle, Pattern matching- The wild-cards, Escaping and Quoting, Redirection: The three standard file, /dev/null and /dev/tty: Two special files, pipes, *tee*: Creating a Tee, Command Substitution, Shell variables.

**UNIT - II**

**FILE ATTRIBUTES**


**THE PROCESS**


**UNIT - III**

**SIMPLE FILTERS**


**FILTERS USING REGULAR EXPRESSIONS – grep AND sed**

Searching for a pattern, Basic regular expressions (BRE) – An Introduction, Extended regular expressions (ERE) and egrep.

The stream editor, Line addressing, Using multiple instructions (-E and -F), Context addressing, Writing selected lines to the file (w), Text editing, Substitution.
UNIT - IV

SHELL PROGRAMMING
Shell scripts, read: Making scripts interactive, Using command line arguments, exit and Exit status of Command, The logical operators && and || - Conditional Execution, The if conditional, Using test and [] to evaluate expressions, The case conditional, expr: Computation and string handling, $0: Calling a script by different names, while: Looping, for: Looping with a list, set and shift: Manipulating the positional parameters, The here document(<<), trap: Interrupting a program, Debugging shell script with –x
Shells and sub-shells, export: Exporting shell variables, Running a script in the current shell: The . command, let computation, Arrays, String handling, Conditional parameter substitution, Merging streams, Shell functions, eval: Evaluating twice, The exec statement. 8 Hrs

UNIT - V

awk – AN ADVANCED FILTER
Simple awk Filtering, Splitting a Line into Fields, printf: Formatting output, Variables and Expressions, The Comparision operators, Number Processing, Variables, The –f option: storing awk programs in a file, The BEGIN and END sections, Built in variables, Arrays, Functions, Control flow,- the if statement, Looping with for, Looping with while.

Perl-THE MASTER MANIPULATOR
Perl preliminaries, The chop function: Removing the last character, Variables and operators, The string handling functions, Specifying filenames in command line, $_: The default variable, Current line number($.) and the Range operator(..), Lists and arrays, foreach: Looping through a list, split: Splitting into a list or array, join: Joining a list, dec2bin.pl:converting a Decimal Number to binary, grep: Searching an array for a pattern, Associative arrays, Regular expressions and substitution, file handling, File tests, Subroutines.

8 Hrs

TEXT BOOKS:

REFERENCE BOOK :
LABORATORY COMPONENTS

<table>
<thead>
<tr>
<th>Hrs/Week</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>01</td>
</tr>
</tbody>
</table>

Total Hours: 26 (13 weeks)

Following is the typical set and in the examination student may be asked to write and execute any program within the syllabus.

1. Programs on Binary trees – Inserting into Binary search tree, tree traversals and other related algorithms.
2. Shell sort, Radix sort, Address calculation sort, Searching, Indexed Sequential search, Hashing Techniques.
3. Programs on File structures: Indexed sequential search, inserting and searching records into a file.
4. Students must familiarize themselves with vi editor on Linux platform. Write some C and C++ programs, compile them and execute. (Programs from Data structures and OOP can be taken as examples). Debugging using debug utility on Linux must be tried.
5. X-windows GUI on Linux platform must be studied.
6. Perl and Awk utilities must be used in programming.
7. Shell programs have to be written.
8. ‘Make’ file utility and other useful utilities must be tried.
9. Students must be trained on Installation of Linux OS.

INDIVIDUAL EFFECTIVENESS LABS (IEL)

Subject Code: 13IS407  Credits: 02
Hrs / Week: 4  Total Hours: 52

Introduction

Entry Edge (E²) is an industry readiness program designed for technology undergraduates to help them enhance important individual behavior & skills, and become productive from the very beginning of their corporate carrier. The program places a high emphasis on the pedagogy of learning by doing.

As part of the program, students first go through individual behavior & skill labs (Individual Effectiveness Labs) in their II year of engineering curriculum and then participate in “hands on” and “minds on” team activities in a simulated work environment, to accomplish
tasks and to solve real-world organizational issues during a week long Immersive Group Workshop (IGW) held in the III year of their engineering course.

This document provides the syllabus and evaluation framework for Individual Effectiveness Labs (IEL).

**Objectives:**

1. To help the students understand themselves. Identify and analyze personality/behavioral attributes of personal effectiveness – exploratory orientation, self-disclosure, receptivity to feedback and sensitivity to others.
2. To help the students identify their primary and secondary motivators – what drives them for achievement?
   a. Understanding the student’s need for achievement
   b. Understanding how positive expectations lead to positive results.
3. To help the students to develop a goal driven mindset and to take the first steps into individual personal planning, controlling and measuring results.
4. To make the students aware of importance of communication and typical barriers to communication.
5. To help the students develop effective oral communication skills.
6. To help the students develop effective written communication skills.
7. To help the students develop listening skills.
8. To help the students participate in group discussions.
9. To help the students develop effective business presentation skills.
10. To help the students receive feedback with an open mind, respond to feedback and take the action on them.
11. To help the students develop time management and organization skills.

**CONTENTS:**

**Module 1: Know Yourself**

Self assessment profilers to identify and assess the following – Identify and analyze personality/behavioral attributes of personal effectiveness – exploratory orientation, self disclosure, receptivity to feedback, sensitivity to others. **8 Hrs**
Module 2: Achievement Motivation & Goal Setting
- Identifying primary and secondary motivators using a motivational profiler.
- Understanding need for achievement.
- Developing goal driven mindset.
- First steps into career planning. 8Hrs

Module 3: Communication Skills
- Effective oral communication
- Effective written communication
- Constructing effective messages (memo, letters, e-mails)
- Writing persuasively
- Correspondence etiquettes – letters & email
- Importance of listening responsively
- Handling conversations
- Effective group discussions 14 Hrs

Module 4: Presentation Skills
- Understanding audience, presentation objectives, best practices & tools in preparation of presentation.
- Improving quality of presentation through better use of voice, eyes, gestures, visual aids.
- Presenting to groups
- Presenting one-on-one. 10 Hrs

Module 5: Handling Feedback
- Seeking feedback
- Accepting feedback with an open mind
- Responding to feedback
- Actionizing feedback 6 Hrs

Module 6: Time Management
- Introduction to Time Management and importance of managing self
- Beating procrastination
- Action plans-starting to achieve in a small way
- Scheduling skills 6 Hrs

Reference Books
2. Online reference materials provided as part of the Entry Edge program.