
B. E. SYLLABUS

COMPUTER SCIENCE & ENGINEERING

III & IV SEMESTER

With
Scheme of Teaching
& Examination
# DEPARTMENT: COMPUTER SCIENCE & ENGINEERING

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Name of the Faculty</th>
<th>Qualification</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Dr. Niranjan.N. Chiplunkar</td>
<td>Ph.D.</td>
<td>Professor/Principal</td>
</tr>
<tr>
<td>2.</td>
<td>Dr. Sarojadevi H.</td>
<td>Ph.D.</td>
<td>Professor and HOD</td>
</tr>
<tr>
<td>3.</td>
<td>Dr. K.R. Uday Kumar Reddy</td>
<td>Ph.D.</td>
<td>Professor</td>
</tr>
<tr>
<td>4.</td>
<td>Dr. Sreekantha D. K.</td>
<td>Ph.D.</td>
<td>Professor</td>
</tr>
<tr>
<td>5.</td>
<td>Dr. Neelima B.</td>
<td>Ph.D.</td>
<td>Professor</td>
</tr>
<tr>
<td>6.</td>
<td>Mrs. Sharada U. Shenoy</td>
<td>M.Tech(Ph.D.)</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>7.</td>
<td>Mr. Venugopala P.S.</td>
<td>M.Tech(Ph.D.)</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>8.</td>
<td>Mr. Roshan Fernandes</td>
<td>M.Tech(Ph.D.)</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>9.</td>
<td>Ms. Shalini P.R.</td>
<td>M.Tech(Ph.D.)</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>10.</td>
<td>Mr. Radhakrishna Dodmane</td>
<td>M.Tech(Ph.D.)</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>11.</td>
<td>Mr. Raju K.</td>
<td>M.Tech(Ph.D.)</td>
<td>Associate Professor</td>
</tr>
<tr>
<td>12.</td>
<td>Mr. Pradeep Kanchan</td>
<td>M.Tech(PhD)</td>
<td>Asst.Prof. Gd III</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fulltime - Sponsored</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Mr. Ravi B.</td>
<td>M.Tech(Ph.D.)</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>15.</td>
<td>Mr. Vijaya Murari T.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd III</td>
</tr>
<tr>
<td>16.</td>
<td>Mr. Chandra Naik</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>17.</td>
<td>Mr. Manjunath Kamath</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>18.</td>
<td>Mrs. Pallavi K. N.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>19.</td>
<td>Mr. Hemanth Kumar G.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>20.</td>
<td>Mr. Pradeep Nazareth</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>21.</td>
<td>Mr. Ranjan Kumar H. S.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>Qualification</td>
<td>Position</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------</td>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>22</td>
<td>Mrs. Anisha P. Rodrigues</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>23</td>
<td>Mrs. Minu P. Abraham</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>24</td>
<td>Mr. Ramesha Shettigar</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>25</td>
<td>Ms. Savitha Shetty</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>26</td>
<td>Ms. Sharmila S. Sequeira</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>27</td>
<td>Mr. Sannidhan M.S.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>28</td>
<td>Mr. Naveen Chandaverkar</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>29</td>
<td>Mr. Pawan Hegde</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>30</td>
<td>Mrs. Keerthana B. C.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>31</td>
<td>Mrs. Asmita Poojari</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>32</td>
<td>Mr. Raghunandan K. R.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd II</td>
</tr>
<tr>
<td>33</td>
<td>Mrs. Shabari Shedthi B.</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>34</td>
<td>Mr. H R Manjunath Prasad</td>
<td>M.Tech(Ph.D.)</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>35</td>
<td>Mr. Krishna Prasad Rao</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>36</td>
<td>Mr. Sunil Kumar Aithal</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>37</td>
<td>Mr. Shashank Shetty</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>38</td>
<td>Mr. Sampath Kini</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>39</td>
<td>Mrs. Divya Jennifer D’Souza</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
<tr>
<td>40</td>
<td>Mr. Mahesh Kini</td>
<td>M.Tech</td>
<td>Asst.Prof. Gd I</td>
</tr>
</tbody>
</table>
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision:
To become a hub of academic activities of the Computer Science and Engineering and a center of excellence in the field of computer science education.

Mission:
To transform the students into computer science graduates who would be ready to take up any challenges in the field of computer science as well informed, attuned, adapted and responsible by imparting the state of the art concepts and technologies.

Programme Educational Objectives (PEOs) :

1. Graduates must gain the ability to identify, formulate & solve challenging Computer science and Engg. Problems both theoretically and practically.
2. Graduates must develop professional and communication skills that prepare them for immediate employment or for adapting to emerging trends by engaging in life-long learning in Computer science and related disciplines.
3. Graduates be provided with an educational foundation that prepares them for leadership roles along diverse career paths and work in a team.
4. Graduates must develop an understanding of the social and human context in which their engineering contribution will be utilized.
Programme Outcomes (POs) :

After successful completion of the BE program in Computer Science and Engineering discipline, the students are expected to have attained the following abilities.

1. An ability to apply knowledge of mathematics, science and engineering.
2. An ability to design and conduct experiments, as well as to analyze and interpret data.
3. 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.
4. An ability to function on multidisciplinary teams.
5. An ability to identify, formulate and solve engineering problems.
6. An understanding of professional and ethical responsibility.
7. An ability to communicate effectively.
8. The broad education necessary to understand the impact of engineering solutions in global, economic environmental and societal context.
9. A recognition of the need for and an ability to engage in lifelong learning.
10. An ability to apply engineering and project management principles.
11. An ability to use the techniques, skills and modern engineering tools necessary for engineering practice.
12. An ability to understand and apply the concepts of programming and computer design & technology.
## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
### SCHEME OF TEACHING AND EXAMINATION

**III SEMESTER B.E.**

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Code</th>
<th>Subject</th>
<th>Theory/Tuto./Prac./ Self Study</th>
<th>Total Hrs./Week</th>
<th>C.I.E.</th>
<th>S.E.E.</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14CS301</td>
<td>Generating Functions and Transform Techniques</td>
<td>4+0+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>14CS302</td>
<td>Logic Design</td>
<td>3+0+0+0</td>
<td>3</td>
<td>50</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>14CS303</td>
<td>Discrete Mathematical Structures</td>
<td>4+0+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>14CS304</td>
<td>Basics of Web Programming</td>
<td>3+2+0+S</td>
<td>5</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>14CS305</td>
<td>Theory and Practice of Data Structures</td>
<td>4+0+2+0</td>
<td>6</td>
<td>50+50</td>
<td>50+50</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>14CS306</td>
<td>Object Oriented Programming</td>
<td>3+0+2+0</td>
<td>5</td>
<td>50+50</td>
<td>50+50</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>14CS307</td>
<td>Logic Design Lab</td>
<td>0+0+2+0</td>
<td>2</td>
<td>50</td>
<td>50</td>
<td>1</td>
</tr>
</tbody>
</table>

**Total** | **29** | **29** | **450** | **450** | **25**
<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Code</th>
<th>Subject</th>
<th>Theory/Tuto./Prac./ Self Study</th>
<th>Total Hrs./Week</th>
<th>C.I.E.</th>
<th>S.E.E.</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14CS401</td>
<td>Probability theory and numerical methods</td>
<td>4+0+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>14CS402</td>
<td>Design &amp; Analysis of Algorithms</td>
<td>3+0+2+0</td>
<td>5</td>
<td>50+50</td>
<td>50+50</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>14CS403</td>
<td>Finite Automata &amp; Formal languages</td>
<td>4+1+0+0</td>
<td>5</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>14CS404</td>
<td>Computer Organization &amp; Architecture</td>
<td>4+0+0+S</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>14CS405</td>
<td>Data Communications</td>
<td>4+0+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>14CS406</td>
<td>Unix Programming</td>
<td>3+0+2+0</td>
<td>5</td>
<td>50+50</td>
<td>50+50</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>14CS407</td>
<td>Individual Effectiveness Labs(IEL)</td>
<td>0+4+0+0</td>
<td>4</td>
<td>50</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td><strong>31</strong></td>
<td><strong>450</strong></td>
<td><strong>450</strong></td>
<td><strong>26</strong></td>
</tr>
</tbody>
</table>
GENERATING FUNCTIONS AND TRANSFORM TECHNIQUES

Sub code : 14CS301/14IS301  Credits : 04
Hrs/Week : 4+0+0+0  Total Hours : 52

Course Outcomes:
At the end of the course the student will be able to

1. Understand the concepts like graph, subgraphs, isomorphic graphs, proper coloring and chromatic polynomial of a graph and their applications.
2. Use Fourier series and harmonic analysis techniques to represent periodic functions in terms of sine or cosine waves and understand the concept of Fourier transforms of functions.
3. Apply ordinary generating functions/ exponential generating functions to solve real life related problems.
4. Solve difference equations with boundary conditions using Z-transforms techniques.
5. Apply derangement/rook polynomial techniques as a tool to solve problems on life related situations.

UNIT - I

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

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. Fourier sine and Fourier cosine transforms, Inverse Fourier sine and Inverse Fourier cosine transforms(simple problems)  12 Hours
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 Hours

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 Hours

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 Hours

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

Sub code : 14CS302  Credits : 03
Hrs/Week : 3+0+0+0  Total Hours : 39

Course Outcomes:
At the end of the course the student will be able to

1. Comprehend and analyze basic logic gates and the digital logic circuit design concepts.
2. Describe ALU design and compare the alternatives.
3. Analyze and design sequential circuits, counters and registers.
4. Describe and analyze advanced digital logic design using HDL, and mixed signal systems.

UNIT-I

Logic gates and combinational logic:


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 simplification, Simplification by Quine-McClusky Method, Hazards and Hazard cover, HDL Implementation Models.

UNIT-II

ALU design

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, Troubleshooting with a Logic Probe, 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, Arithmetic Circuits using HDL. **8 Hours**

**UNIT-III**

**Basics of Sequential circuits**

**Flip-Flops**: RS Flip-flops, Gated Flip-flops, Edge-triggered RS, D, JK Flip-flops, Flip-flop timing, JK Master-slave Flip-flops, Switch Contact Bounce Circuits, Various Representations of Flip-flops, Analysis of Sequential Circuits, Conversion of Flip-flops – a synthesis example, HDL implementation of Flip-flop. Design of Sequential Circuit: Design and Analysis of Synchronous and Asynchronous Sequential Circuits with examples. **8 Hours**

**UNIT-IV**

**Registers and Counters**

**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. **8 Hours**

**UNIT-V**

**Mixed signal systems and IC families**


**Digital Integrated Circuits**: CMOS inverter, CMOS NAND gate, CMOS NOR gate, Two input TTL NAND gate, TTL NOR gate, AND-OR-INVERT gate. **7 Hours**
TEXT BOOK:

REFERENCE BOOKS:

Mode of evaluation: Continuous evaluation tests, quizzes, assignments

***********************

DISCRETE MATHEMATICAL STRUCTURES

Sub code : 14CS303
Credits : 04
Hrs/Week : 4+0+0+0
Total Hours : 39

Course Outcomes:
At the end of the course the student will be able to
1. Understand sets, relations, functions, posets, and lattice.
2. Design and formulate recursive formula.
3. Deduce and infer logical statements.
4. Properties of integers and prime numbers.

UNIT-I

Sets and Relations

Set Theory and Counting.
Introduction to Set, Operations on sets, Principle of inclusion exclusion, The Pigeonhole principle and Recurrence Relations.

Relations and its Properties.
Product sets and Partitions, Relations and Digraphs, Paths in relations and Digraphs. 8 Hours
UNIT-II

More about relations and Functions

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. 8 Hours

UNIT-III

Relations, structures and Lattices:

Order relations and Structures
Partially Ordered Sets, External elements of Partially ordered sets, Lattices, Introduction to Group theory, Semi groups, Groups and Abelian groups. 7 Hours

UNIT-IV

Logic fundamentals

Fundamentals of logic.
Propositions and Logical Operations, Conditional Statements, Biconditional statements, The laws of logic, Logical implication-Rules of inference, Quantifiers, Mathematical induction. 8 Hours

UNIT-V

Number theory

Introduction to number Theory :
Applications of this subject to the area of Computer Science is to be discussed in the class. **8 Hours**

**TEXT BOOKS:**


**REFERENCE BOOKS:**


**Mode of evaluation:** Continuous assessment tests, assignments, tutorial sheets etc.

**************************

**BASICS OF WEB PROGRAMMING**

<table>
<thead>
<tr>
<th>Sub code</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>14CS304</td>
<td>04</td>
</tr>
<tr>
<td>Hrs/Week</td>
<td>Total Hours</td>
</tr>
<tr>
<td>3+2+0+S*</td>
<td>42</td>
</tr>
</tbody>
</table>

*Self Study to be exercised under the supervision of course instructor and to be restricted to not more than 10% of the total teaching hours.

**Course Outcomes:**
At the end of the course the student will be able to
1. Understand basic concepts of PHP, Java Script and learn the constructs of these languages.
2. Design and analyze working of forms and databases.
3. Implement and apply various constructs learnt.

Lab component are to be taught as tutorials.

UNIT – I

Chapter 1 : Introduction to HTML
HTML Documents - Dividing the document into 2 parts(Headers tags Body tags), Paragraphs, Formattings, Elements of an HTML Document - Text Elements ,Tag Elements, Special Character elements, Image tags, HTML Table tags, Lists(Numbered list, Non-Numbered lists, Definition lists), Anchor tag, Name tag , Hyperlinks - FTP/HTTP/HTTPS, Links with images and buttons, Links to send email messages, Text fonts and styles, background colors/images, Marquee Behavior, Forms related tags (action, method,name,input,submit etc), Lab components

8 Hours

UNIT – II

Chapter 2 : Form Handling
Introduction, Creating Forms in HTML,GET and POST, Accessing Form data, $_POST, $_GET, $_REQUEST, Handling the file upload, Saving the uploaded file, Restricting the file type/size, Checking for errors, File inclusion.

Chapter 3 : JavaScript
Introduction JavaScript , HTML DOM, JavaScript Data type, Loops in JavaScript, Functions in JavaScript, Embedding JavaScript in HTML, Lab components

8 Hours

UNIT – III

Chapter 4 : PHP Basics
Introduction to PHP , Support for Database, PHP Installation, Working with PHP, Why PHP?, Basic Syntax of PHP, PHP statement terminator and case insensitivity, Embedding PHP in HTML, Comments, Variables, Assigning value to a variable, Constants, Managing Variables.

Chapter 5 : Operators

**Chapter 6 : Functions**
Functions in PHP, User-Defined function, Function Definition, Function Call, Function with arguments, Function with return value, Call by value and call by references, Understanding variable scope, Global Variables, Static Variables, Include and Require, Built-in functions in PHP

9 Hours

**UNIT – IV**

**Chapter 7 : Arrays**
Introduction to Array, Array in PHP, Creating an Array, Accessing Elements of an Array, Modifying Elements of an Array, Finding the Size of an Array, Printing an Array in the readable way, Iterating Array Elements, Modifying Array while iteration, Iterating Array with Numeric index, Removing Element from an Array, Converting an Array to String, Converting String to an Array, Array Sorting, Multidimensional Array, Accessing elements of a Multidimensional Array, Iterating Multidimensional Array.

**Chapter 8 : PHP File Handling**
Introduction, File Open, File Creation, Writing to files, Reading from File, Searching a record from a file, Closing a File, Using PHP With HTML Forms.

8 Hours

**UNIT – V**

**Chapter 9 : Class and Object**
Introduction, Object, Class, Defining Class in PHP, Object in PHP, Usage of $this variable, Constructor, Constructor with Parameters.

**Chapter 10 : Exception Handling**
Introduction to Exception, Exception Handling mechanisms, Creating Custom Exceptions, Multiple Catch Blocks, Exception Propagation, Error Handling in PHP.

**Chapter 11:Database and MYSQL**
What is Database?, MYSQL, SQL, SQL Functions, PHP and MYSQL, Execute Queries

9 Hours

TEXT BOOKS:

1. PHP Bible - Tim Converse
2. PHP A beginners guide - Bill McCarthy
3. PHP and MySQL Web Development - Luke Welling

LABORATORY COMPONENT (Tutorial)

Basics Programming
- Exercise 1 – Branching Statements using character
- Exercise 2 - Branching Statements using number
- Exercise 3 – Looping Statement
- Exercise 4 – String Functions
- Exercise 5 – String Manipulation
- Exercise 6 - Calculator
- Exercise 7 - Strings

Practicals using Functions
- Exercise 8 – Generate Employee ID
- Exercise 9 – Calculate Tax
- Exercise 10 – Reverse a string
- Exercise 11 – Call by value and Call by reference
- Exercise 12 – Find Grade

Practicals using Arrays
- Exercise 13 – Sorting
- Exercise 14 – Find grade
- Exercise 15 – Sort Array
- Exercise 16 – Multidimensional Array
- Exercise 17 – Population Details

File Handling programs
- Exercise 18 – Writing into a existing file
- Exercise 19 – Read from a file
- Exercise 20 – Filter the contents from the file
• Exercise 21 – File Copy

**PHP programming through HTML**
• Exercise 22 – PHP with HTML

**Programs related with PHP Classes and Objects**
• Exercise 23 – Student Registration
• Exercise 24 – Online Examination System
• Exercise 25 – Online Feedback System

**Exception Handling in PHP**
• Exercise 26 – User Defined Exception
• Exercise 27 – Exception Propagation
• Exercise 28 – Error Handling in PHP

**Java Scripting**
• Exercise 29 – Arithmetic Operation
• Exercise 30 – Html and java script

**************

**THEORY & PRACTICE OF DATA STRUCTURES**

**Sub Code : 14CS305**

**Credits : 04**

**Hrs/Week : 4+0+2+0**

**Course Outcomes:**
At the end of the course the student will be able to
1. Understand basic data types.
2. Design and analyze linear data structures such as stacks, queues and linked lists.
3. Implement and apply and various data structures.
4. Analyze hierarchical data structures such as trees and graphs.

**UNIT-I**

**Introduction And Overview**
Definitions, Concepts of data structures, types, Overview of data structures.

**Pointers** Definition and Concepts, Accessing variables through pointers, Pointers and functions, Arrays and pointers, Array of pointers, Pointer
arithmetic.

Linear Data Structures - Stacks
Introduction and Definition, Representation of stack: Array and structure representation of stacks, Operations on stacks, Applications of stack: Conversion of Expressions, Evaluation of expressions, Recursion: Implementation, Simulating Recursion, examples on Recursion. 10 Hours

UNIT - II

Linear Data Structures - Queues
Introduction and Definition Representation of Queue: Array and Structure representation of queue, various queue Structures: ordinary queue, circular queue, priority queue.

Linear Data Structures - Singly Linked lists
Memory allocation functions. Definition and concepts singly Linked List: Representation of link list in memory, Operations on singly Linked List, Circular Linked List 10 Hours

UNIT-III

Linear Data Structures - Doubly Linked lists

Nonlinear Data Structures - Tree data structures 1

UNIT-IV

Nonlinear Data Structures - Tree data structures 2
Expression Tree: Evaluating expression tree, Constructing expression tree from postfix expression, traversals, Threaded binary Tree: types, B-Trees, B+ Trees, AVL Trees: Definition, Constructing a general AVL tree. 10 Hours

UNIT-V

Nonlinear Data Structures - Graphs
Graph terminologies: Walks, Paths, Circuits, Connected graphs, Disconnected graphs and Components, Euler graphs. Directed graphs,
Undirected graphs, Hamiltonian paths and Circuits.

**Representation of graphs**
Set Representation, Linked representation, Matrix representation. Operations on Graphs: Insertion and Deletion of edges and vertices (linked representation), DFS, BFS.  

**TEXT BOOKS:**

**REFERENCE BOOKS:**
1. *Classic Data Structures*; D.Samanta.

***********

**DATA STRUCTURES LAB**

Sub Code : 14CS305  
Credits : 01  
Hrs/Week : 2

**Course Outcomes:**
At the end of the course the student will be able to

1. Understand basic data types.
2. Design and analyze linear data structures such as stacks, queues and linked lists.
3. Implement and apply and various data structures.
4. Analyze hierarchical data structures such as trees and graphs.

Data Structures Programs using C

Contains programs on data structures such as stacks, queues, linked lists and trees and operations on them. Programs will be implemented using C.

************

OBJECT ORIENTED PROGRAMMING

<table>
<thead>
<tr>
<th>Sub code</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>14CS306</td>
<td>03</td>
</tr>
<tr>
<td>Hrs/Week</td>
<td>3+0+2+0</td>
</tr>
<tr>
<td>Total Hours</td>
<td>39</td>
</tr>
</tbody>
</table>

Course Outcomes:
At the end of the course the student will be able to
1. Understand basic concepts of object oriented programming
2. Design and analyze concepts of classes and objects
3. Implement and apply object oriented approaches like inheritance, polymorphism, operator overloading and function overloading
4. Analyze the working of files, generic programming using templates, and exception handling

UNIT-I
Overview of Object Orientation & Basics of C++

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, Applications 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.  

UNIT-II

Classes and Objects

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 Overloading

Constructors and Destructors
Introduction, Constructors, Parameterised 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 and Polymorphism

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. 8 Hours

UNIT-V

Exceptions and File handling

Templates and Exception Handling
Introduction, Function Templates, Class Templates, Overloading of Template Functions.
Basics of Exception Handling, Exception Handling Mechanism, Limitation of Exception Handling. 8 Hours

Working with files
Classes for Files Stream Operations, Opening and Closing a File, Error Handling during File Operations.

TEXT BOOK:
1. E.Balagurusamy: Object – Oriented Programming with C++, Third Edition, Tata McGraw Hill. (Chapters 1.3 to 1.8, 2.1, 2.2, 2.6, 3.5 to 3.7, 3.10 to 3.18, 4.3, 4.6 to 4.9, 5, 6, 7, 8, 9, 11, and 12)

REFERENCES BOOKS:
2. Herbert Schildt: C++ The Complete Reference
3. K.R. Venugopal: Mastering C++

***************
OBJECT ORIENTED PROGRAMMING LAB

Sub code : 14CS306
Credits : 1
Hours/Week : 3+0+2+0

Students have to write, execute and test programs IN C++ covering the syllabus of 14CS306. Typical example programs are:
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

LOGIC DESIGN LABORATORY

Sub code : 14CS307
Credits : 1
Hours/Week : 0+0+2+0

1. To study and verify the truth table of logic gates
2. Realization of a given Boolean function using Basic gates and Universal gates
3. Realization of Basic gates using Universal gates.
4. Prove and implement DeMorgans theorem
5. Design and implementation of a Half-adder and a Full-adder using minimum number of 2-input NAND gates
6. Design and implementation of adder/subtractor circuit using IC7483
7. To design and realize the following using IC 7483.
   a) BCD to Excess-3 Code
   b) Excess-3 to BCD Code.
8. Given any four variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.

9. Design and implementation of a Full adder and a Full subtractor using 3:8 decoder and 4 input NAND gates.

10. Realization of one & two bit comparator circuit using IC7485

11. To set up and test a 7-segment static display system to display numbers 0 to 9.

12. Design and implementation of the following using 4-bit shift register
   a. Ring counter
   b. Johnson counter

13. Design and implement following flipflops:
   a. RS Flipflop(Set and Reset)
   b. D Flipflop(DATA)
   c. T Flipflop(Toggle)

14. Design and implement following flipflops:
   a. JK Flipflop
   b. JK Master-Slave(JK MS) flipflop.

15. Design and implementation of a Mod-N (N<8) Synchronous up counter using J-K flip flop ICs.

**Using Verilog/VHDL, simulate the following:**

1. Code and simulate all Basic gates(any 5)
2. Simulation of 8:1 multiplexer.
3. Simulation of full adder.
4. Simulation of the following
   i. Multiplexer
   ii. Demultiplexer

5. Simulation of the following:

a. Ring counter
b. Johnson counter

***************
PROBABILITY THEORY AND NUMERICAL METHODS

Sub code : 14CS401/ IS401/CV401  Credits : 04
Hrs/Week : 4+0+0+0  Total Hours : 52

Course Outcomes:
At the end of the course the student will be able to

1. Understand and appreciate probabilistic models for situations involving chance effect
2. Identify if there exists any relationship between two given variables by applying the concepts of correlation and regression.
3. Understand the concepts of random samples, sampling distribution and central limit theorem
4. Apply interpolation, numerical differentiation and integration methods to solve engineering problems
5. Solve initial value problems, algebraic and transcendental equations using various numerical methods.

UNIT - I
Introduction to probability: finite sample space, conditional probability and independence. Baye’s theorem(Overview). One dimensional random variable, pdf, cdf, expectation and variance. Two and higher dimensional random variables, joint pdf, marginal pdf, covariance, correlation coefficient. 12 Hours

UNIT - II
Distributions: Binomial, Poisson, uniform, normal, Gamma, exponential & Chi- square Simple problems.
Sampling Theory-Random samples, sampling distribution, t and F distributions correlation and regression. 12 Hours

UNIT - III
Stochastic Process: Definition, Classifications, Types of Stochastic processes, Markov process, Markov Chains, Transition probabilities, Higher transition probabilities, C-K equations. 8 Hours
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 Hours**

UNIT - V


**TEXT BOOKS:**


**REFERENCE BOOKS:**

Mehdi: Stochastic Proces


***************

**DESIGN & ANALYSIS OF ALGORITHMS**

<table>
<thead>
<tr>
<th>Sub code</th>
<th>14CS402</th>
<th>Credits</th>
<th>03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/Week</td>
<td>3+0+2+0</td>
<td>Total Hours</td>
<td>52</td>
</tr>
</tbody>
</table>

**Course Outcomes:**

At the end of the course the student will be able to
1. Understand basics of algorithms and problem solving using algorithms
2. Learn mathematical analysis of algorithmic efficiency
3. Design and develop algorithms of various types such as searching, sorting, divide & conquer, decrease and conquer, transform & conquer etc. for problems solving
4. Know space-time tradeoff and dynamic programming
5. Work on problems using greedy techniques

UNIT - I

Introduction

Introduction: What is an Algorithm?, Fundamentals of Algorithmic, Problem Solving, Important Problem Types, Fundamental Data Structures

UNIT - II

Brute Force and Divide & Conquer

Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search
Divide and Conquer: Mergesort, Quicksort, Binary Search, Binary tree traversals and related properties, Multiplication of large integers and Stressen’s Matrix Multiplication.

UNIT - III

Decrease & Conquer and Transform & Conquer

Decrease & Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Algorithms for Generating Combinatorial Objects

UNIT-IV

Tradeoffs and Dynamic Programming

Time and Space Tradeoffs: Sorting by Counting, Input Enhancement in String Matching, Hashing

UNIT-V

Greedy, Backtracking and Branch & Bound Techniques

Greedy Method: 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.

TEXT BOOK:

REFERENCE BOOKS:
DESIGN & ANALYSIS OF ALGORITHMS LABORATORY

Sub code : 14CS402  
Credits : 01

Students have to write, execute and test programs covering the syllabus of 14CS402. The execution times 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. Strassen matrix multiplication
5. Heapsort
6. Prim’s algorithm
7. Kruskal’s algorithm
8. Dijkstra’s algorithm
9. Warshal’s algorithm
10. Floyd algorithm
11. Travelling salesperson problem
12. N-Queen problem

Note: Students may implement the programs using C/C++ language on Windows platform.

******************
FINITE AUTOMATA AND FORMAL LANGUAGES

Sub code : 14CS403 Credits : 04
Hrs/Week : 4+0+0+0 Total Hours : 52

Course Outcomes:
At the end of the course the student will be able to

1. Explain the basic concepts of deterministic and non-deterministic finite automata, regular language, context-free language, Turing machines, halting problem, computability and complexity
2. Analyze, and design finite automata, pushdown automata, Turing machines, formal languages, and grammars
3. Demonstrate the knowledge of regular expression and regular language
4. Solve the problems using formal language and apply it in compiler design

UNIT - I

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

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.

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-examples. Right linear grammar generate regular languages, Right linear grammars for regular languages.
Properties of Regular Languages
Closure properties of regular languages-Closure under simple set operations, closure under other operations. Identifying nonregular languages-Using the Pigeonhole principle, A pumping lemma. **5 Hours**

UNIT-III

Context-Free languages: Context-Free grammars-examples, Leftmost and rightmost derivations, Derivations trees, Parsing and ambiguity-Ambiguity in grammars and languages. **5 Hours**

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. **5 Hours**

UNIT - IV

Properties of Context-Free Languages.
Two pumping lemmas:A pumping lemma for Context-Free languages,A pumping lemma for Linear languages. **5 Hours**

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. **5 Hours**

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. **7 Hours**
A Hierarchy of Formal Languages and Automata:
Recursive and recursively enumerable languages, The Chomsky hierarchy. 3 Hours

Limits of Algorithmic Computation:
Some problems that cannot be solved by Turing machines, The post correspondence problem. 2 Hours

TEXT BOOK:

REFERENCE BOOKS:

***************

COMPUTER ORGANIZATION AND ARCHITECTURE
Sub code : 14CS404
Credits : 04
Hrs/Week : 4+0+0+S*
Total Hours : 52

*Self Study to be exercised under the supervision of course instructor and to be restricted to not more than 10% of the total teaching hours.

Course Outcomes:
At the end of the course the student will be able to
1. Understand organization of computer, its component parts, structural design and connectivity
2. Understand the operation of the arithmetic unit and carry out arithmetic operations
3. Describe different ways of communication withI/O devices and standard I/O interfaces
4. Explain the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit
5. Explain the hierarchical memory system including cache memories and virtual memory

UNIT - I

Basic Computer Organization

Basic structure of computer and its components, Machine Instructions and Programs, Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Stacks and Queues, Subroutines, Additional Instructions, Number representation and operation on numbers in IEEE format. Comparison of RISC and CISC architectures.  

11 Hours

UNIT - II

Arithmetic Operations


10 Hours

UNIT - III

Processing Unit

Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control, Microprogrammed Control, Pipelining basics.  

10 Hours

UNIT-IV

Memory Systems

Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories –Mapping Functions, FIFO and LRU replacement policies, Performance
Considerations, Virtual Memories, Secondary Storage.  

10 Hours

UNIT-V

Input/ Output Organization


11 Hours

TEXT BOOKS:


REFERENCE BOOKS:


************************

DATA COMMUNICATIONS

Sub code : 14CS405
Credits : 04
Hrs/Week : 4+0+0+0
Total Hours: 52

Course Outcomes:
At the end of the course the student will be able to
1. Understand the basics of networking
2. Understand the concepts of protocol standard and network models.
3. Describe difference between analog and digital transmission concepts.
4. Explain the digital transmission concepts and error detection logics
5. Explain the concepts of circuit switching and data link concepts.

UNIT-I

Introduction to Data Communications


UNIT-II

Digital Transmission Fundamentals - I

Analog and Digital Data, Analog and Digital Signals, Transmission impairment Data rate limits: Niquist Bit rate and Shannon Channel

**UNIT-III**

**Digital Transmission Fundamentals – II**


Transmission Media: Twisted Pair, Coaxial Cable, Fiber Optic Cable, Radio waves, Microwaves, Infrared

Error Detection and Correction: Introduction, Block Coding, Linear Codes, Cyclic Codes, Cyclic redundancy check, Hardware implementation, Cyclic code analysis, Internet Checksum. **10 Hours**

**UNIT-IV**

**Switching and Multiplexing**

**CIRCUIT-SWITCHED NETWORKS:** Three Phases, Efficiency, Delay, Circuit-Switched Technology in Telephone Networks.

**DATAGRAM NETWORKS:** Routing Table Datagram Networks in the Internet, VIRTUAL-CIRCUIT NETWORKS, STRUCTURE OF A SWITCH

Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Synchronous Time-Division Multiplexing, Statistical Time-Division Multiplexing, SPREAD SPECTRUM: Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum. **11 Hours**
UNIT-V

Data link control and multiple access

RANDOM ACCESS, ALOHA, CONTROLLED ACCESS, CHANNELIZATION 11 Hours

TEXT BOOK:

REFERENCE BOOKS:
1. Alberto Leon Garcia and Indra Widjaja, Communication Networks – Fundamental Concepts and Key architectures, 2nd edition

******************
UNIX PROGRAMMING

Sub code: 14CS406
Hrs/Week : 3+0+2+0

Course Outcomes:
At the end of the course the student will be able to
1. Learn various Unix commands and work using them
2. Understand Shell process and learn Shell script programming
3. Learn Awk & PERL scripts and develop programs using these scripts
4. Do the advanced shell programming using export, eval, exec etc.

UNIT-I
Unix Operating System, The UNIX architecture and command usage, The File System, Basic file Attributes, the vi Editor, The Shell and basic commands and their usage 8 Hours

UNIT-II
More file Attributes, Simple filters, Filters using regular expressions.
8 Hours

UNIT-III
The process, Essential Shell Programming.
8 Hours

UNIT-IV
Perl-The Master Manipulator.
Awk –An Advanced filter.
8 Hours

UNIT-V
Advanced Shell programming.
7 Hours
TEXT BOOKS:
3. Python Cookbook, Alex Martelli, Anna Ravenscroft, David Ascher, O'Reilly Media Publication, Second Edition

REFERENCE BOOK:

************

UNIX PROGRAMMING LAB

Sub code : 14CS406             Credits : 1

1. 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.
2. X-windows GUI on Linux platform must be studied.
3. Perl and awk utilities must be used in programming.
4. Shell programs have to be written.
5. ‘Make’ file utility and other useful utilities must be tried.
6. Students must be trained on Installation of Linux OS

************

INDIVIDUAL EFFECTIVENESS LABS (IEL)

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).

***************

INDIVIDUAL EFFECTIVENESS LABS (IEL)

Sub Code: 14CS407          Credits: 02
Hrs / Week: 0+4+0+0

Course Outcomes:
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 Hours

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. 8 Hours

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 15 Hours

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.  13 Hours

**Module 5: Handling Feedback**
• Seeking feedback
• Accepting feedback with an open mind
• Responding to feedback
• Actionizing feedback  6 Hours

**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 Hours

**REFERENCE BOOKS:**
2. Online reference materials provided as part of the Entry Edge program.