B. E. SYLLABUS

COMPUTER SCIENCE & ENGINEERING

III & IV SEMESTER

With

Scheme of Teaching

& Examination

DEPARTMENT: COMPUTER SCIENCE & ENGINEERING

Sl. No.	Faculty Name	Qualification	Designation
1.	Dr. Niranjan.N. Chiplunkar	Ph.D	Principal
2.	Dr. K R Udaya Kumar Reddy	Ph.D	Professor & Head
3.	Dr. Udaya Kumar K Shenoy	Ph.D	Professor
4.	Dr. Deviprasad M	Ph.D	Professor
5.	Dr. D.K. Sreekantha	Ph.D	Professor
6.	Dr. Seetharam K	Ph.D	Professor
7.	Mrs. Jyothi Shetty	M.Tech, (Ph.D)	Asso. Prof
8.	Mrs. Sharada Udaya Shenoy	M.Tech, (Ph.D)	Asso. Prof
9.	Mr.Venugopala P.S.	M.Tech, (Ph.D)	Asso. Prof
10.	Mr. RoshanFernandes	M.Tech, (Ph.D)	Asso. Prof
11.	Mr. Radhakrishna	M.Tech, (Ph.D)	Asso. Prof
12.	Mrs. Sarika Hegde	M.Tech, (Ph.D)	Asso. Prof
13.	Mr. Raju K	M.Tech, (Ph.D)	Asso. Prof
14.	Mr. Sudeepa K.B	M.E., (Ph.D)	Asso. Prof
15.	Mr. Pradeep Kanchan	M.Tech, (Ph.D)	Asst. Prof Gd III
16.	Mr. Ravi B	M.Tech, (Ph.D)	Asst. Prof Gd III
17.	Mr. VijayaMurari T	M.Tech, (Ph.D)	Asst. Prof Gd III
18.	Mrs. Pallavi KN	M.Tech(Ph.D)	Asst. Prof Gd II
19.	Mr. Ranjan Kumar HS	M.Tech	Asst. Prof Gd II
20.	Mrs. Anisha P Rodrigues	M.Tech, (Ph.D)	Asst. Prof Gd II
21.	Mr. Raghunandan KR	M.Tech	Asst. Prof Gd II
22.	Mrs. Minu P. Abraham	M.Tech	Asst. Prof Gd II
23.	Mr. Ramesha Shettigar	M.Tech	Asst. Prof Gd II
24.	Mr. SampathKini	M.Tech	Asst. Prof Gd II
25.	Mr. Mahesh Kini. M	M.Tech	Asst. Prof Gd II
26.	Mrs. Asmita Poojary	M.Tech	Asst. Prof Gd II
27.	Mrs. Shruthi M	M.Tech, (Ph.D)	Asst. Prof Gd II

28.	Ms. Savitha	M.Tech	Asst. Prof Gd I
29.	Mr. Sannidhan M.S	M.Tech	Asst. Prof Gd I
30.	Mr. SunilkumarAithal	M.Tech	Asst. Prof Gd I
31.	Mrs. Keerthana B. Chigateri	M.Tech	Asst. Prof Gd I
32.	Mr. Pawan Hegde	M.Tech	Asst. Prof Gd I
33.	Mrs. ShabariShedthi. B	M.Tech, (Ph.D)	Asst. Prof Gd I
34.	Mr. Naveen Chandawerkar	M.Tech, (Ph.D)	Asst. Prof Gd I
35.	Mr. Krishna Prasad Rao	M.Tech	Asst. Prof Gd I
36.	Mr. Shashank Shetty	M.Tech	Asst. Prof Gd I
37.	Mrs. Divya Jennifer D'Souza	M.Tech	Asst. Prof Gd I
38.	Mr. Puneeth R.P	M.Tech	Asst. Prof Gd I
39.	Mrs. Shilpa Karegoudar	M.Tech	Asst. Prof Gd I
40.	Mrs. Rajalaxmi Prabhu	M.Tech	Asst. Prof Gd I
41.	Dr. Mohammed Javed	Ph.D	Asst. Prof Gd I
42.	Dr. Aravinda C V	Ph.D	Asst. Prof Gd III
43.	Mr. Sandeep Kumar Hegde	M.Tech	Asst. Prof Gd I
44.	Ms. Ankitha A Nayak	M.Tech	Asst. Prof Gd I
45.	Mrs. Swathi Pai M	M.Tech	Asst. Prof Gd I
46.	Ms. Rajashree	M.Tech	Asst. Prof Gd I

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION:

To be a center of excellence in Computer science & Engineering education and research, empower the lives of individuals to fulfill their academic excellence, professional passions, and partnership for community development..

MISSION :

- To impart both theoretical and practical knowledge through the state-of-the-art concepts and technologies in Computer Science and Engineering.
- To inculcate values of professional ethics, leadership qualities and lifelong learning.
- To create professionals for employment in industry, research, higher education, and entrepreneurship to benefit the society.

Programme Educational Objectives (PEOs):

1. Graduates will be capable of practicing principles of Computer Science & Engineering, Mathematics and Engineering sciences to solve problems that are appropriate to the discipline
2. Graduates will be able to contribute to their profession and society
3. Graduates will be employed in computing profession or engaged in learning to pursue higher education

Programme Outcomes (POs):

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

- Apply the knowledge of engineering science and mathematics in solving problems that are appropriate to the discipline
- Apply the knowledge of computing both hardware and software aspects to the solution of real-world engineering problems in the discipline
- Design & develop algorithms, programs, and projects using various software and modern tools appropriate to software industry or Research & Development activities in the discipline

Graduate Attributes :

Sl. No.	Graduate Attributes
a	Engineering Knowledge
b	Problem Analysis
c	Design / development of solutions
d	Conduct investigations of complex problems
e	Modern tool usage
f	The engineer and society
g	Environment and sustainability
h	Ethics
i	Individual and team work
j	Communication
k	Project management and finance
l	Life-long learning

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHEME OF TEACHING AND EXAMINATION

III Semester B.E **30 Hours/Week**

Sl. No.	Sub. Code	Subject	Theory/Tuto./Prac./Self Study	Total Hrs./Week	C.I.E	S.E.E	Credits
1	16CS301	Generating Functions and Transform Techniques	4+0+0+0	4	50	50	4
2	16CS302	Digital Systems Design	3+0+3+0	6	50+50	50+50	5
3	16CS303	Discrete Mathematics	4+1+0+0	5	50	50	4
4	16CS304	Computer Organization & Architecture	4+0+0+0	4	50	50	4
5	16CS305	Data Structures	4+0+3+0	7	50+50	50+50	6
6	16CS306	Data Communications	4+0+0+S	4	50	50	4
TOTAL			**30**	**30**	**400**	**400**	**27**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHEME OF TEACHING AND EXAMINATION

IV Semester B.E **23 Hours/Week**

Sl. No.	Sub. Code	Subject	Theory/Tuto./Prac./Self Study	Total Hrs./Week	C.I.E	S.E.E	Credits
1	16CS401	Linear Algebra and Probability Theory	4+0+0+0	4	50	50	4
2	16CS402	Design & Analysis of Algorithms	4+0+3+0	7	50+50	50+50	6
3	16CS403	Programming with Java	3+1+0+0	4	50	50	3
4	16CS404	Principles and Practices of Software Engineering	4+0+0+S	4	50	50	4
5	16CS405	Microprocessors & Peripherals	4+0+3+0	7	50+50	50+50	6
6	16HU411	Enhancing Self Competence	2+1+0+0	3	50	50	2
TOTAL			**23**	**23**	**400**	**400**	**25**

NOTE: Fast track course on Object Oriented concepts using C++ during beginning of 4th Semester

GENERATING FUNCTIONS AND TRANSFORM TECHNIQUES

Sub Code : 16CS301 **Credits : 04**
Hrs/Week: 4+0+0+0 **Total Hours : 52**

Course Learning Objectives:

This Course will enable students to

1. Identify various types of graphs and apply these concepts in life related problems.
2. Represent periodic functions in the analytic form using Fourier principles.
3. Use various concepts of Z-transform of a function to solve life related problems.
4. Give the difference between nonnegative and positive integer solutions of a linear equation and find the corresponding solutions.
5. Find the sequence generated by exponential generating function.

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

COMBINATORICS

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**

Course Outcomes:

At the end of the course the student will be able to

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

Mapping of POs & COs:

P Os COs	a	b	c	d	e	f	g	h	i	j	k	l
1	M											
2	M				L							
3					H							
4					H						M	
5					H						M	

L: Low M: Medium H : High

TEXT BOOKS:

1. Kreyszig, "Advanced Engineering Mathematics", John Wiley and Sons, VI-Edition.
2. Ralph P.Grimaldi, "Discrete and combinatorial Mathematics", 5th Edition, PHI/Pearson Education, 2004.
3. Narsing Deo, "Graph theory with Applications to Engg. and Computer Science",PHI.
4. Graham knuth&Patashnik, "Concrete Mathematics"

REFERENCE BOOKS:

1. Richard A.Brualdi,"Introductory Combinatorics", 4th Edition, Pearson Prentice Hall, 2004.
2. F.Harary, "Graph theory", Narosa Publishing House, 1988.

****

DIGITAL SYSTEM DESIGN

Sub Code	**: 16CS302**	**Credits**	**: 03**
Hrs/Week	**: 3+0+3+0**	**Total Hours**	**: 39**

Course Learning Objectives:

This course will enable students to:

1. Get an idea of Combinational Logic circuits and its Applications, Simplification of Algebraic Equations using Karnaugh Maps and Quine McClusky Techniques.
2. Describe and Design Digital multiplexers, Decoders, Encoders, Adders and Sub-tractors, Binary comparators.
3. Learn to Analyze and Design different Latches and Master-Slave Flip-Flops.
4. Describe, Design and Analyze Synchronous and Asynchronous counters.
5. Analyze advanced digital logic design using HDL.

UNIT – I

The Basic Gates: Review of Basic Logic gates, Positive and Negative Logic
Combinational Logic Circuits: Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs,Quads, and Octets, Karnaugh Map Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method.
HDL Using VERILOG: Introduction, HDL Implementation Models.

13 Hours

UNIT – II

Data-Processing Circuits: Multiplexers, De-multiplexers, **Decoders**-1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, **Encoders**, **Exclusive-OR Gates**, Parity Generators and Checkers, Magnitude Comparator.
HDL Using VERILOG: HDL Implementation of Data Processing Circuits
Arithmetic Circuits: 2's Compliment Arithmetic, Arithmetic Building Blocks, The Adder Subtractor, Fast Adder,
HDL Using VERILOG: HDL Implementation of Arithmetic Circuits **13 Hours**

UNIT – III

Flip- Flops: RS Flip-Flops, T Flip flop, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIPFLOPs.
Flip- Flops: JK Master-slave FLIP-FLOP, Various Representation of FLIP-FLOPs, Design of synchronous sequential circuits using Moore and Mealy models.
HDL Using VERILOG: 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, Shift Register, Applications of Shift Registers,
Counters: Asynchronous Counters, Synchronous Counters, Counter Design as a Synthesis problem **13 Hours**

Course Outcomes:
Upon completion of this course, students will be able to:

CO1 **: Acquire** the knowledge of combinational logic and the simplification techniques for reduction of logic expressions using karnaugh map and quine-mcklusky methods
CO2: **Apply** the simplification techniques to **describe** various logic circuits
CO3: **Describe** the operation and **design** of various data processing circuits such as multiplexers,demultiplexers, decoders, encoders etc
CO4: **Acquire** the knowledge on the working of adder subtractor circuits and to **analyze** their performance for the simplification of binary numbers
CO5: **Analyze and design** sequential circuits, counters and registers

Graduate Attributes (GA)

This course will map the following GA as per NBA:

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

TEXT BOOK:

1. Donald P Leach, Albert Paul Malvino & Goutam Saha: Digital Principles and Applications, 8th Edition, Tata McGraw Hill, 2015**.**

REFERENCE BOOKS:

1. Stephen Brown, Zvonko Vranesic: Fundamentals of Digital Logic Design with VHDL, 2nd Edition, Tata McGraw Hill, 2005.
2. R D Sudhaker Samuel: Illustrative Approach to Logic Design, Sanguine-Pearson, 2010.
3. M Morris Mano: Digital Logic and Computer Design, 10th Edition, Pearson, 2008.

E-Books / Online Resources:

Online course material

1. https://drive.google.com/drive/folders/0B6YWHA6PoUtmcEQ3bHVaRzYtc00 by M Morris Mano
2. https://sites.google.com/site/dldcse241/ematerials/ebooks
3. http://freevideolectures.com/Course/2319/Digital-Systems-Design from Prof. D. Roychoudhury, IIT- Karagpur

MOOC:

1. Universitat Autònoma de Barcelona: Digital Systems: From Logic Gates to Processors (Coursera)
2. Stanford University: Michael Genesereth, Associate Professor, Introduction to Logic.

DIGITAL SYSTEMS DESIGN LABORATORY

Sub Code : 16CS302 **Credits : 02**
Hrs/ Week : 0+0+3+0

Course Learning Objectives:

This Course will enable students to

1. **Create** logic circuits using the help of truth tables of basic gates such as AND, OR, NOT and universal logic gates such as NAND, NOR.
2. **Apply** Boolean laws and theorems to simplify complex logic circuits and **Design** the circuit.
3. **Design** and **Build** different types of Multiplexers, De-multiplexers, Decoders, Flip Flops, counters
4. **Interpret** binary addition, subtraction, multiplication division and XOR function on binary numbers using simplified circuits .
5. **Construct** various types of Registers in Hardware Definition Language (HDL/VHDL).
6. **Analyze** different types of Digital Integrated Circuits.

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 fallowing flipflops:
    a. RS Flipflop(Set and Reset)
    b. D Flipflop(DATA)

c. T Flipflop(Toggle)

14. Design and implement fallowing 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

Course Outcomes:

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

1. **Design and Implement** various basic logic gates and the digital logic circuit
2. **Design and Implement** combinational Circuits, counters and registers
3. **Design and Implement** Sequential Circuits
4. **Analyze and simulate** HDL programming using Verilog.

****

DISCRETE MATHEMATICS

Sub Code	**: 16CS303**	**Credits**	**: 04**
Hrs/Week	**: 4+1+0+0**	**Total Hours**	**: 50**

Course Learning Objectives:

This course will enable students to:

1. Prepare the background in abstraction, notation, and critical thinking of mathematics related to computer science.
2. Learn the set theoretic concepts and basics of relations.
3. Study about functions and partial ordered sets with applications to computer science.
4. Describe the basics of graph theory, counting techniques and their applications in computer science.
5. Know the relevance of elementary number theory in cryptography.
6. Study the core idea of mathematical induction, recursive relations, logic and their use in computer science.

UNIT – I

Set Theory:
Basics of Sets, Set Operations, Computer Representation of Sets,
Relations and its Properties :
Product sets, Relations and Digraphs, Paths in Relations and Digraphs, Properties of Relations, Partitions, Equivalence Relations, Transitive Closure and Warshall's Algorithm, Computer representation of Relations and Digraphs. **10 Hours**

UNIT – II

Functions:
Definition, Types of Functions, Invertible Functions, Permutation Functions, Functions for Computer Science.
Order relations and Structures: Partially Ordered Sets, Extremal Elements of Partially Ordered Sets, Lattice: Definition and Examples. **10 Hours**

UNIT – III

Graphs and their Applications: Introduction to Graphs, Graph Models, Bipartite Graphs, Modeling Applications using Bipartite Graphs, Representing Graphs: Adjacency Matrices, Incidence Matrices. Some Applications of Special Types of Graphs, Applications of Vertex and Edge Connectivity, Applications of Euler Paths and Circuits, Applications of Hamilton Circuits, Applications of Planar Graphs, Applications of Graph Colorings.
Counting:
Principle of Inclusion-Exclusion, The Pigeonhole Principle, Generalized Pigeonhole Principle, Some Elegant Applications of the Pigeonhole Principle. **10 Hours**

UNIT – IV

Introduction to Number Theory:
Divisibility and The Division Algorithm, The Euclidean Algorithm, Modular Arithmetic, The Extended Euclidean Algorithm, Prime Numbers. Fermat's, and Euler's Theorems, Testing for Primality: Miller-Rabin Algorithm, The Chinese Remainder Theorem, Applications of Congruences: Hashing Functions, Pseudorandom Numbers. **10 Hours**

UNIT – V

Mathematical Induction and Recursion
Mathematical Induction, Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation. Modeling With Recurrence Relations: Rabbits and the Fibonacci Numbers, The Tower of Hanoi, Codeword Enumeration. Divide-and-Conquer Recurrence Relations: Binary Search, Finding the Maximum and Minimum of a Sequence, Fast Multiplication of Integers. Methods of Proof: Direct, Indirect and Proof by Contradiction.
Fundamentals of logic:
Basic Connectives and Truth Tables, The Laws of Logic, Rules of Inference. **10 Hours**

Course Outcomes:

Upon completion of this course, students will be able to:

1. Apply the operations of sets and use Venn diagrams to solve applied problems in Computer Science. Apply relation properties in subjects like Relational database, and Data mining.
2. Apply the properties of functions to application problems and use partial ordered set, lattices in problems related to design of network models.
3. Apply the graph theory concepts in algorithms related to computer networks and use counting principle in solving simple problems and making arguments in computer science.
4. Apply the concepts of number in designing cryptographic algorithms.
5. Apply the recurrence relations in problems related to biology, computer science algorithm and digital signal processing. Use the fundamentals of logic in formal verification, and automated reasoning for various problems.

Graduate Attributes (GA)

This course will map the following GA as per NBA:

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Conduct investigations of complex problems
5. Ethics
6. Life-long Learning

TEXT BOOKS:

1. ”Discrete Mathematics and its applications” , Kenneth H. Rosen, Tata McGraw Hill, 5th Edition-2003.
2. “Discrete Mathematical Structures”, Bernard Kolman, Robert C. Busby, Sharon Ross. 3rd Edition, PHI 2001.
3. “Cryptography and Network Security” , William Stallings, 5th Edition, Pearson Education Asia.

REFERENCE BOOKS:

1. Discrete and Combinatorial Mathematics” By Ralph P. Grimaldi, Pearson Education, Asia, IV Edition-2002.
2. ”Discrete Mathematical Structures with applications to computer Science” By J. P. Tremblay, R. Manohar, Tata McGraw Hill-1987.
3. ”Elementary Number Theory” By David M.Burton, II Edition, UBS NewDelhi.
4. Discrete Mathematics, J K Sharma, 3rd edition, 2013, Macmillan India Ltd.

E-Books / Online Resources:

1. Discrete Mathematics with Algorithms byM. O. Albertson, J. P. Hutchinson - J. 1988, Wiley.
2. Discrete Mathematics for Computer Science, Gary Haggard, John Schlipf, Sue Whitesides, Thomson Brooks/Cole, 2006.
3. http://ocw.mit.edu/courses/mathematics/ (online course material)

MOOC:

1. http:www.nptelvideos.in/2012/11/discrete-mathematical-structures.html
2. https: www.khanacademy.org
3. http: www.cs.berkeley.edu/~daw/teaching/cs70-s05/

****

COMPUTER ORGANIZATION AND ARCHITECTURE

Sub Code : 16CS304		**Credits**	**: 04**
Hrs/Week : 4+0+0+0		**Total Hours**	**: 50**

Course Learning Objectives:

This course will enable students to:

1. **Outline** the basic structure and operation of a digital computer
2. **Learn** about arithmetic unit and **perform** fixed point and floating point addition, subtraction, multiplication and division in binary 2's complement number system
3. **Appreciate** the fine grain details of basic processing unit in terms of control unit, arithmetic and logical unit, memory unit and I/O unit
4. **Remember and comprehend** the hierarchical memory system including cache memories and virtual memory
5. **Tell how** different ways of communication with I/O devices and standard I/O interfaces

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- Subroutine nesting and processor stack, parameter passing, *Comparison of RISC and CISC architectures (self study)*.

10 Hours

UNIT – II

ARITHMETIC OPERATIONS

Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division, Floating-point Numbers and Operations on numbers in IEEE format.

10 Hours

UNIT – III

BASIC PROCESSING UNIT

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

Instruction level Parallelism- What is Pipelining? (**Text book-2** Appendix A -A.1), Pipeline Hazard- Structural, Data , and Control hazards (Appendix A -A.2).

Thread level Parallelism- Moore's law, Motivation for Multi-core processors, Overview of a typical dual core architecture, Introduction to thread level parallelism. (**Text book-3** Chapter-1) **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 (Self study)*

10 Hours

UNIT – V

INPUT/OUTPUT ORGANIZAT

Input / Output Organization: Accessing I/O Devices, Interrupts –Interrupt Hardware, Enabling and Disabling Interrupts, Exceptions, Handling Multiple Devices, Controlling Device Requests, Buses, Direct Memory Access, Interface Circuits (parallel, Serial). *Standard I/O Interfaces – PCI Bus, SCSI Bus, USB (Self study- Basics only)*

10 Hours

Course Outcomes:

Upon completion of this course, students will be able to:

1. **Discuss** organization of computer, its component parts, structural design and connectivity.
2. **Practice** the arithmetic operations performed by ALU.
3. **Comprehend** the basic structure of processors, and modern trends in processor technology.
4. **Explain** the structure of various memory systems including cache memories and virtual memory.
5. **Portray** the design of basic and standard I/O interfaces.

Graduate Attributes (GA)

This course will map the following GA as per NBA:

1. Engineering Knowledge
2. Design/Development of Solutions
3. Problem Analysis
4. Life-long Learning

TEXT BOOKS:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky **,** Computer Organization, , 5th Edition, TMH, 2002.
2. John L. Hennessey and David A. Patterson**,** Computer Architecture, A Quantitative Approach, 4th Edition, Elsevier, 2007
3. Shameem Akhter and Jason Roberts, Multicore programming- Increasing performance through software multithreading, Intel press, 2006

REFERENCE BOOKS:

1. William Stallings, **Computer Organization & Architecture**, 7th Edition, PHI, 2006.
2. Vincent P. Heuring & Harry F. Jordan, **Computer Systems Design and Architecture**, 2nd Edition, Pearson Education, 2004.
3. David A. Patterson, John L. Hennessy, "**Computer Organization and Design**", 4th Edition *Elsevier, 2012.*
4. John P.Hayes, **Computer Architecture**, 2nd edition, McGraw Hill, 1988.

E-Books / Online Resources:

1. dcs.abu.edu.ng/staff/sani-ahmad-hassan/course materials/COSC303_LEC.pdf
2. http://www.cse.iitm.ac.in/~vplab/courses/comp_org/
3. http://www.ddegjust.ac.in/studymaterial/msc-cs/ms-07.pdf
4. http://nsec.sjtu.edu.cn/data/MK.Computer.Organization.and.Design.4th.Edition.Oct.2011.pdf

MOOC:

1. http://nptel.ac.in/courses/106103068/
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-823-computer-system-architecture-fall-2005/syllabus/

****

DATA STRUCTURES

Sub Code : 16CS305		**Credits**	**: 04**
Hrs/Week : 4+0+3+0		**Total Hours**	**: 50**

Course Learning Objectives:

This course will enable students to:

1. **Outline** the concepts of data structures, its types, pointers and stacks.
2. **Get the idea of** linear data structures such as queue, singly linked lists and circular linked lists.
3. **Implement the operations of** doubly linked list and binary trees.
4. **Identify** the concepts of different types of tree data structures.
5. **Make use of** graph data structure, its representation and applications.

UNIT – I

Introduction: Data Structure, Classification (Primitive and non primitive), data structure operations.

Pointers: Definition and Concepts, Accessing variables through pointers, Pointers and functions, Arrays and pointers, Array of pointers.

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
Dynamic 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
Doubly Linked List: Representation and Operations, Circular doubly Link list: Representation and Operations. Linked List representation of stack, Linked List representation of queue.
Nonlinear Data Structures - Tree data structures-1
Basic Terminologies, Binary Trees: Properties, Representation of Binary Tree: Linear representation, Linked representation, Operations on Binary Tree: Insertion, Simple Deletion, Traversals. Introduction to Binary Search Tree. **10 Hours**

UNIT – IV

N**onlinear Data Structures - Tree data structures-2**
Expression Tree: Constructing expression tree from postfix expression, traversals, Application of tree: **Evaluation of expression, programming examples**
Threaded binary Tree: types, B-Trees, B+ Trees, AVL Trees: Definition, Constructing a general AVL tree. **10 Hours**

UNIT – V

Nonlinear Data Structures - Graphs
Introduction to Graph terminologies
Representation of graphs
Set Representation, Linked representation, Matrix representation. Operations on Graphs: Insertion and Deletion of edges and vertices (linked representation), Depth First Search, Breadth First search.
Hashing: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing.
10 Hours

Course Outcomes:

1. **Acquire** knowledge of various types of data structures, pointers and stack.
2. **Apply** the concept of data structures such as queue, singly linked lists and circular linked lists for solving problems
3. **Design** various functions for implementation of doubly linked lists and binary trees.
4. **Implement and apply** the concept of tree data structure.
5. **Identify** the different representations and applications of graph data structure.

Graduate Attributes (GA)

This course will map the following GA as per NBA:

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

TEXT BOOKS:

1. Fundamentals of Data Structures in C – Ellis Horowitz and Sartaj Sahni, 2nd edition, Universities Press, 2014.
2. Data Structures – Seymour Lipschutz, Schaum's Outlines, Revised 1st edition, McGraw Hill, 2014.

REFERENCE BOOKS:

1. Data Structure using C; Aaron M. Tenenbaum, Yedidyah Langsam & Moshe J. Augenstein; Pearson Education/PHI, 2006.
2. Data Structures Using C And C++; 2nd Edition; Langsam Yedidyah, Augenstein Moshe J., Tenenbaum Aaron M; Prentice-Hall 2009.

E-Books / Online Resources:

1. Data Structures Using C, ISRD Group, Tata McGraw Hill, 2006
2. Data Structures Using C, Reema Thareja, 2nd edition, Oxford University Press, 2014

MOOC:

1. Introduction to Data Structures by edx , URL : https://www.edx.org/course/
2.Data structures by Berkley, URLl: https://people.eecs.berkeley
3. Advance Data Structures by MIT OCW , URL: https://www.mooclab.club/
4. Data Structure by Harvard Extension School, URL: http://www.extension.harvard.

****

DATA STRUCTURES LABORATORY

Sub Code	**: 16CS305**	**Credits**	**: 02**
Hrs/Week	**: 0+0+3+0**	**Total Hours**	**: 39**

Course Learning Objectives:

This Course will enable students to

1. **Demonstrate** linear data structures like stack, queue .Apply stack data structure implementing postfix evaluation, infix to postfix conversion and Tower of Hanoi problem.
2. **Implement** different types of linked list and its applications.
3. **Implement** non linear data structures like binary trees and graphs.

Students have to write, execute and test programs covering the syllabus of 15CS305.

Typical problems that may be tried are

1. Stack static implementation.
2. Queue static implementation.
3. Application of stack data structure.
4. Different types of queue.
5. Tower of Hanoi problem using recursion.
6. Singly Linked list implementation.
7. Dynamic implementation of stack data structure.
8. Dynamic implementation of queue data structure.
9. Circular linked list implementation.
10. Doubly linked list and Circular doubly linked list implementation.
11. Binary Tree Construction and Tree traversal operations.
12. Construction of Binary Search Tree and Postfix Expression tree .

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

Course Outcomes:

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

1. Design and implement abstract data a type such as stack and queue and it is applications**.**
2. Design and Implement various **l**inear data structures like linked list and its different types.
3. Design and Implement various non linear data structures like binary trees and graph.

****

DATA COMMUNICATIONS

Sub Code : 16CS306	**Credits : 04**
Hrs/Week : 4+0+0+S	**Total Hours : 50**

*** 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 Learning Objectives:

This course will enable students to:

1. Outline the theory behind the basic design of networks and approaches to design networks.
2. Get the idea of representation of digital information and digital transmission and could able to understand and design reliable transmission.
3. Able to determine type of transmission and technology required for transmission.
4. Outline the theory behind the various protocol used in transmission and could design new protocols.
5. Get the idea of how to access the transmission medium based on various protocols and determine new design.

UNIT – I

Communication Networks and Services - Networks and Services, Approaches to network design.

Applications and Layered Architecture – Examples of layering, OSI reference model, Overview of TCP/IP architecture. **10 Hours**

UNIT – II

Digital Transmission Fundamental – Analog and Digital Data, Analog and Digital Signals, Transmission impairment, Data rate limits: Nyquist Bit rate and Shannon Channel Capacity, Performance.

Digital Transmission: Digital to digital conversion, Line coding - RZ, NRZ, Manchester, Differential Manchester and AMI, Block coding- 4B/5B.

Error detection and correction: Hamming code, CRC, Internet Checksum.

10 Hours

UNIT – III

Analog Transmission: Digital to analog conversion - Amplitude shift keying, Frequency shift keying and phase shift keying. Quadrature Amplitude shift keying.

Transmission Systems: Multiplexing, SONET, Wavelength division multiplexing.

10 Hours

UNIT – IV

Peer-To-Peer Networks: Peer – to – Peer protocols and service models, ARQ protocols, Other adaptation functions.

Medium Access Control Protocols: Multiple access communications, Scheduling approaches to MAC, Channelization. **10 Hours**

UNIT – V

Local Area Networks – LAN Protocols, IEEE 802.3 standard – frame structure, physical layer , Fast Ethernet, Gigabit Ethernet 802.11 standard – frame structure and addressing, medium Access Control. **10 Hours**

Course Outcomes:

Upon completion of this course, students will be able to:

1. Explain the basics of networking, Network Models.
2. Identify the data & signals, the channel rate and conversion techniques, the error detection and correction techniques.
3. Demonstrate handling of multiple channels and transmission.
4. Determine how to transmit data over the channels reliably.
5. Describe the Data link control and multiple access methods.

Graduate Attributes (GA)

This course will map the following GA as per NBA:

1. Engineering Knowledge
2. Design/Development of Solutions
3. Problem Analysis

TEXT BOOKS:

1. Alberto Leon-Garcia & Indra Widjaja, "Communication Networks - Fundamental Concepts and Key Architectures", 2nd edition, Tata McGraw Hill.
2. Behrouz A.*Forouzan*, "Data Communications and Networking", 5th edition, Tata McGraw Hill.

REFERENCE BOOKS:

1. William Stallings, Data and Computer Communication, 5th Edition, Prentice Hall India.
2. William A. Shay, Understanding Data Communications and Networks, 2nd Edition, Thomson
3. Godbole, Data Communications and Networks, Tata McGraw-Hill 2002
4. fghfgh Micael A. Gallo & William M. Handcock, Computer Communications and Networking Technologies, 2003 Edition, homson.

E-Books / Online Resources:

1. http://eng.uok.ac.ir/abdollahpouri/Network/A.Leon-Garcia_Communication_Networks.pdf
2. https://vtucsenotes.wordpress.com/2014/02/23/computer-networks/
3. http://highered.mheducation.com/sites/0073376221/student_view0/index.html

MOOC:

1. http://nptel.ac.in/downloads/106105080/

2. http://www.nptelvideos.in/2012/11/data-communication.html
3. http://nptel.ac.in/courses/106105082/
4. https://www.youtube.com/playlist?list=PL374944B232C0B48E
5. http://freevideolectures.com/Course/2278/Data-Communication

****

LINEAR ALGEBRA AND PROBABILITY THEORY

Sub Code : 16CS401 **Credits : 04**
Hrs/Week : 4+0+0+0 **Total Hours : 50**

Course Learning Objectives:

This Course will enable students to

1. **Apply** the theory of matrices and vector spaces to solve various engineering problems.
2. **Understand** the notion of linear transformation, Gradient and Hessian of linear and quadratic function.
3. **Solve** problems using Bayes' theorem and find marginal distribution and variance of two dimensional random variable.
4. **Derive** the mean and variance of Binomial, Poisson and normal distributions.
5. **Classify** a given situation and solve life related problems in stochastic process. **State** the importance of sampling distribution.

UNIT – I

MATRIX & VECTOR SPACE:
Review of Matrix Properties, Trace, Norms, Relation between trace and Eigen values of a matrix, Eigen values and Eigen vectors of symmetric matrices.
Vector spaces, subspaces, bases and dimension, coordinates, Summary of row-equivalence and computations concerning subspaces. 10 Hours

UNIT - II

LINEAR TRANSFORMATIONS:
Linear transformations, algebra of linear transformations, representation of transformations by matrices, isomorphism, Range and Null space.
Gradient, Hessian, Gradient and Hessian of linear and quadratic functions, Least squares. 10 Hours

UNIT – III

INTRODUCTION TO PROBABILITY:
Finite sample space, conditional probability and independence(overview),Bayes' theorem. One dimensional random variable, pdf , cdf , expectation and variance. Two and higher dimensional random variables, joint pdf and marginal pdf. **10 Hours**

UNIT - IV

DISTRIBUTIONS: Binomial, Poisson,uniform,normal,gamma and exponential distributions, simple problems. Curve fitting(linear and quadratic), correlation and regression.

10 Hours

UNIT – V

STOCHASTIC PROCESS:
Definition, Classifications, Types of Stochastic processes, Markov process, Markov Chains, Transition probabilities, Higher transition probabilities, C-K equations**.**

SAMPLING THEORY: Random samples, sampling distribution, t, F and Chi-squrae distributions.

10 Hours

Course Outcomes:

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

1. **Find** trace, eigen values and eigen vectors of the given symmetric matrix. Realize the importance of the notions of basis and dimension in the study of vector spaces.
2. **Demonstrate** the concept of linear transformation as a linear function from one vector space to another. **obtain** the Gradient and Hessian of linear transformation.
3. **Classify** and **appreciate** probabilistic models for situations involving chance effect and appreciate the concepts of pdf, cdf, random variables and its consequences.
4. **Illustrate** some of the important distributions of discrete random variables and continuous random variables. **Apply** the concepts of correlation and regression in real life situations.
5. **State** and **apply** the concepts of Markov processes and transition probabilities in various engineering problems. **Apply** the concepts of sampling in real life situations.

Mapping of POs & COs:

P Os COs	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
1	**M**				**H**							
2					**L**						**M**	
3					**M**							
4		**H**										
5											**H**	

H: High M: Medium L: Low

TEXT BOOKS:

1. B.S.Grewal, “Higher Engineering Mathematics” – 36th Edition.
2. P.L.Meyer, “Introduction of Probability and Statistical Applications” , second Edn. 1975, American Publishing.

3. David C Lay, "Linear Algebra and its applications", 3rd Edition, Person Education (Asia) Pvt. Ltd., 2005.
4. Sean Dineen, "Multivariate Calculus and Geometry", Second Edition, Springer Publications.

REFERENCE BOOKS:

1. Hogg and Craig, "Introduction of Mathematical Statistics", 4th Edn 1975 McMillan.
2. J.Medhi, "Stochastic Process".
3. B. V. Ramana, "Higher Engineering Mathematics", Tata Mc Graw –Hill Ltd.

****

DESIGN & ANALYSIS OF ALGORITHMS

Sub Code : 16CS402 **Credits : 04**
Hrs/Week : 4+0+3+0 **Total Hours : 50**

Course Learning Objectives:

This course will enable students to:

1. Analyze the non-recursive and recursive algorithms and to represent Efficiency of these algorithms in terms of the standard Asymptotic notations.
2. Learn the Brute Force and Divide and Conquer techniques to design the algorithms.
3. Analyze the Decrease and Conquer and Transform and Conquer algorithm design techniques.
4. Get the idea of Time and Space Tradeoffs in designing algorithms and also get the idea of dynamic programming.
5. Get the idea of Greedy, Backtracking and Branch and Bound algorithm design techniques.

UNIT – I

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

Fundamentals of the algorithms Efficiency:
Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms – Fibonacci Numbers and Tower of Hanoi. **10 Hours**

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.

10 Hours

UNIT – III

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

Transform and Conquer: Presorting, Balanced Search Trees, Heaps and Heap sort, Problem Reduction. **10 Hours**

UNIT – IV

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

Dynamic Programming: Computing a Binomial Coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem and memory functions. **10 Hours**

UNIT – V

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. **10 Hours**

Course Outcomes:

Upon completion of this course, students will be able to:

1. Analyze non-recursive or recursive algorithm and to represent in terms of standard Asymptotic notations.
2. Apply Brute Force or Divide and Conquer algorithm design techniques to a given real time problem.
3. Apply the Decrease and Conquer and Transform and Conquer algorithm design techniques to a given real time problem.
4. Analyze Time and Space Tradeoffs in designing algorithms and also to apply dynamic programming to a given real time problem.
5. Apply Greedy, Backtracking and Branch and Bound algorithm design techniques to real time problems.

Graduate Attributes (GA):

This course will map the following GA as per NBA:

1. Engineering Knowledge
2. Design/Development of Solutions
3. Problem Analysis
4. Life-long Learning

TEXT BOOKS:

1. Introduction to the Design & Analysis of Algorithms, Anany Levitin, 2nd Edition, Pearson Education, 2007.

REFERENCE BOOKS:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E.Leiserson, Ronal L. Rivest, Clifford Stein, 2nd Edition, PHI, 2006.
2. Computer Algorithms by Horowitz E., Sahni S., Rajasekaran S,Galgotia Publications, 2001.
3. Introduction to the Design and Analysis of Algorithms A Strategic Approach, R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T.Tsai, TMH, 2005.

E-Books / Online Resources:

- Online course material: http://www.facweb.iitkgp.ernet.in/~sourav/daa.html

MOOC:

- **NPTEL:** http://nptel.ac.in/courses/106101060/

****

DESIGN & ANALYSIS OF ALGORITHMS – LAB

Sub Code : 16CS402 **Credits : 2**
Hrs/Week : 0+0+3+0

Laboratory Experiments: In the laboratory the students must implement the following experiments in Java programming language:

- Various Sorting/Searching algorithms
- Graph traversals – DFS and BFS, Connectivity and Reachability of graphs
- Topological Sorting
- Descending Priority Queue using Heap
- Horspool string matching algorithm
- Binomial coefficient, Warshall's algorithm, Floyd's algorithm, Knapsack problem using Dynamic

Programming and by using memory functions.

- Prim's, Kruskal's, Dijkstra's algorithms
- N-Queens problem.

****

PROGRAMMING WITH JAVA

Sub Code : 16CS403	**Credits : 03**
Hrs/Week : 3+1+0+0	**Total Hours : 39**

Course Learning Objectives:

This course will enable students to:

1. Learn fundamental features of object oriented language and JAVA programming constructs.
2. Develop and run simple Java programs using OOPS concepts of java
3. Create multi-threaded programs and event driven Graphical User Interface (GUI) programming using swing package.
4. Create user interface using swings.

UNIT – I

Introduction to Java: Java's magic: the Byte code; Java Development Kit (JDK); the Java Buzzwords, Object-oriented programming; Simple Java programs. Data types, variables and arrays, Operators, Control Statements.

Classes, Inheritance: Classes fundamentals; Declaring objects; Call by value and Call by Reference, array of objects, Constructors, this keyword, and usage of static keyword.
Inheritance: inheritance basics, using super, creating multi-level hierarchy, method overriding, abstract classes, final classes. **13 Hours**

UNIT – II

Exception handling, packages and interfaces: Exception handling in Java, use of try, catch blocks, multiple catch blocks, finally block, use of throw and throws clauses, creating custom exceptions. Packages, Access Protection, Importing Packages, Interfaces.
Multi-Threaded Programming:
What are threads? How to make the classes threadable; Extending threads; Implementing runnable interface; creating multiple threads, join and isAlive methods of Thread class, Thread Synchronization; achieving thread synchronization among multiple threads. Thread priorities, methods to get and set thread priority. **13 Hours**

UNIT – III

Event Handling: Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model;
Swings:
The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon;JTextField;The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; Jtable. **13 Hours**

Course Outcomes:

Upon completion of this course, students will be able to:

1. Explain the object-oriented concepts and JAVA programming features
2. Develop computer programs to solve real world problems in Java.
3. Create multi threaded programs in Java
4. Create packages and handle the exceptions in Java programming.
5. Develop simple GUI interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles using swings.

Course Outcomes:

Upon completion of this course, students will be able to:

1. Explain the object-oriented concepts and JAVA programming features
2. Develop computer programs to solve real world problems in Java.
3. Develop simple GUI interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles using swings.

Graduate Attributes (GA)

This course will map the following GA as per NBA:

1. Design/Development of Solutions
2. Problem Analysis
3. Modern tool usage

TEXT BOOK:

1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.

REFERENCE BOOKS:

1. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008, ISBN:9788131720806
2. Rajkumar Buyya,S Thamarasi selvi, xingchen chu, Object oriented Programming with Java, Tata McGraw Hill education private limited.
3. Richard A Johnson, Introduction to Java Programming and OOAD, CENGAGE Learning.
4. E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies.

E-Books / Online Resources:

1. Online course material by Oracle :
    a. http://docs.oracle.com/javase/tutorial/index.html
2. https://www.udemy.com/courses/search/?q=java&price=price-free&view=grid

MOOC:

1. Oracle: *www.oracle.com/events/global/en/java.../java-a-beginners-guide-1720064.pdf*
2. NPTEL:*www.nptelvideos.com/java/java_video_lectures_tutorials.php*

****

PRINCIPLES AND PRACTICES OF SOFTWARE ENGINEERING

Sub Code : 16CS404	**Credits : 04**
Hrs/Week : 4+0+0+S	**Total Hours : 50**

*** 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 Learning Objectives:

This course will enable students to:

1. Outline software engineering principles and activities involved in building large software programs.
2. Identify ethical and professional issues and explain why they are of concern to software engineers.
3. Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation.
4. Recognize the importance of software maintenance and describe the intricacies involved in software evolution.
5. Apply estimation techniques, schedule project activities and compute pricing.

UNIT – I

Introduction: Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics, Case Studies. Software Processes: Models: Waterfall Model Incremental Model and Spiral Model Process activities.

Requirements Engineering: Requirements Engineering Processes, Requirements Elicitation and Analysis, Functional and non-functional requirements, The software Requirements, Document Requirements, Specification Requirements, and validation Requirements.

10 Hours

UNIT – II

System Models: Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering.

Design and Implementation: Introduction to RUP, Design Principles, Object-oriented design using the UML, Design patterns, Implementation issues, Open source development.

10 Hours

UNIT – III

Software Testing: Development testing, Test-driven development, Release testing, User testing, Test Automation.

Software Evolution: Evolution processes, Program evolution dynamics, Software maintenance, Legacy system management. **10 Hours**

UNIT – IV

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

UNIT – V

Agile Software Development: Coping with Change, The Agile Manifesto: Values and Principles, Agile methods and Extreme Programming, Plan-driven and agile development, Agile project management, Scaling agile methods. **10 Hours**

Course Outcomes:

Upon completion of this course, students will be able to:

1. Design a software system, component, or process to meet desired needs within realistic constraints.
2. Practice professional and ethical responsibility in software development.
3. Function on multi-disciplinary teams of software project.
4. Use the techniques, skills, and modern engineering tools necessary for engineering practice
5. Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems

Graduate Attributes (GA)

This course will map the following GA as per NBA:

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

TEXT BOOKS:

1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012.

REFERENCE BOOKS:

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

E-Books / Online Resources:

1. http://agilemanifesto.org/
2. **http://www.jamesshore.com/Agile-Book/**

MOOC:

1. UML Class Diagrams for Software Engineering by edX https://www.mooc-list.com/course/uml-class-diagrams-software-engineering-edx
2. Enterprise Software Lifecycle Management by edX https://www.mooc-list.com/course/enterprise-software-lifecycle-management-edx
3. Mastering the Software Engineering Interview by Coursera https://www.mooc-list.com/course/mastering-software-engineering-interview-coursera

****

MICROPROCESSORS AND PERIPHERALS

Sub Code : 16CS405 **Credits : 04**
Hrs/Week : 4+0+0+0 **Total Hours : 50**

Course Learning Objectives:

This Course will enable students to:

1. **Outline** the internal architecture of 8086 microprocessor, concept of addressing modes, instruction set and develop and execute basic programs.
2. **Develop and Execute** modular assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
3. **Interface** microprocessor toexternal I/O devices namely logic controller, stepper motor, seven segment display, DAC, and keypad.
4. **Generate** machine code for 8086 instructionsand **Outline** the working and benefits of 8259A Priority Interrupt Controller.
5. **Understand** the hardware components of 8086 microprocessorand comparethe salient features of advanced microprocessors.

UNIT – I

8086 internal architecture: The Programming Model, Multipurpose registers, Special purpose Registers, Segment registers. Real mode memory addressing, Protected mode memory addressing, Flat mode addressing.

8086 Addressing modes: Register addressing, Immediate addressing, Direct addressing, Register Indirect addressing, Based addressing with displacement, Indexed addressing with displacement, Based Indexed addressing, Based Indexed addressing with displacement.

8086 Instruction Set - 1: Data transfer instructions (including I/O transfers), Binary arithmetic instructions, Decimal (BCD, ASCII) arithmetic instructions, Logical instructions, Shift and rotate instructions, Control transfer instructions.

8086 programming based on Instruction Set - 1: Programs based on data transfer instructions, binary arithmetic instructions, logical instructions, shift and rotate instructions, control transfer instructions.

10 Hours

UNIT – II

Modular Programming: Using procedures, Using macros, Comparison between procedure and macro.

Data Conversions: ASCII to BCD/Hexadecimal, BCD/Hexadecimal to ASCII, 8 bit BCD to Hexadecimal, 8 bit Hexadecimal to BCD.

Using the Key Board and Video Display: DOS & BIOS interrupts, Disk files, Example Programs.

8086 Assembler Directives: ASSUME, DB, DW, DD, END, ENDP, ENDS, EQU, EVEN, EXTRN, GLOBAL, GROUP, INCLUDE, OFFSET, PROC, PUBLIC, SEGMENT, MACRO, ENDM.

10 Hours

UNIT – III

8086 Instruction Set - 2: String instructions, Flag control instructions, Miscellaneous instructions. Programs based on String instructions.

Digital Interfacing: Methods of parallel data transfer,8255A internal block diagram and system connections,8255A operational modes and initialization, Constructing and sending 8255A control words. (Refer Text Book 2).

Interfacing 8086 microprocessor to Logic Controller Interface, Seven Segment Display Interface, Stepper Motor Interface, DAC Interface, and Keypad Interface.

10 Hours

UNIT – IV

8086 Instruction Format (MOV instruction only): Generating machine code for register to register data transfer, memory/register to register/memory data transfer, immediate data transfer, segment register data transfer.

Combining Assembly Language with C/C++: Using Assembly Language with C/C++ for 16-Bit DOS Applications and 32-Bit Applications.

8086 Interrupts: 8086 Interrupts and Interrupt responses, 8086 Interrupt types, 8259A Priority Interrupt Controller – 8259A overview and system connections and cascading, Initializing an 8259A including Initialization Command Words (ICW) and Operational Command Words (OCW). (Refer Text Book 2). **10 Hours**

UNIT – V

8086 Hardware Specifications: 8086 pin functions, Bus buffering and latching (8086 only), Simplified 8086 write and read bus cycle, Minimum mode versus Maximum mode.

Introduction to Pentium Microprocessors: Pentium memory system, Branch prediction logic, Cache structure, Superscalar architecture, Special Pentium registers, Pentium memory management.

Comparison: Comparison of all Intel microprocessors in terms of clock frequency, Register size, Data bus size, maximum address space. (Refer Text Book 2, Page Number 16.13).

10 Hours

Course Outcomes

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

- **Analyze** the internal architecture of 8086 microprocessor, concept of addressing modes and instruction assembling and to develop simple 8086 programs.
- **Develop and Execute** modular assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
- **Interface** microprocessor to external I/O devices namely logic controller, stepper motor, seven segment display, DAC, keypad, and elevator.
- **Generate** machine code for 8086 and understand the working and benefits of 8259A Priority Interrupt Controller.

- **Explain** the hardware components of 8086 and **analyze** the salient features of advanced microprocessors.

TEXT BOOKS:

1. Barry B Brey: The Intel Microprocessors, 8th Edition, Pearson Education, 2009.
2. Douglas V Hall: Microprocessors and Interfacing, Tata McGraw-Hill Publication, Second Edition, 2006.

REFERENCE BOOKS:

1. Liu & Gibson: Microcomputer Systems: The 8086/8088 Family Architecture, Programming and Design, PHI, 2006.
2. Carl Hamachar, Zvonko Vranesic, Safwat Zaky: Computer Organization, Tata McGraw-Hill Publication, 2001.

MOOC:

NPTEL: http://nptel.ac.in/courses/108107029/39

****

MICROPROCESSORS AND PERIPHERALS LABORATORY

Sub Code : 16CS405 **Credits : 02**
Hrs/Week : 0+0+3+0

Course Learning Objectives:

This Course will enable students to

1. **Write and Execute** assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
2. **Interface** microprocessor to external I/O devices like logic controller, stepper motor, seven segment display, DAC, keypad, elevator.

3. **Implement** Real Mode programming in Pentium.

Students have to write, execute and test programs covering the syllabus of 13CS503.
Typical problems that may be tried are:

1. Searching
2. Sorting
3. String manipulation
4. usage of Macros and subroutines
5. DOS interrupt usage
6. BIOS interrupt usage
7. Keyboard interface
8. Display interface
9. Logic controller interface
10. DAC interface
11. Stepper Motor control interface
12. Elevator interface
13. Some examples of Real Mode programming in Pentium.

Course Outcomes:

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

1. **Develop and Execute** assembly level language program for 8086 and must be able to write assembly level program for any processor by studying its architecture.
2. **Interface** microprocessor to external I/O devices like logic controller, stepper motor, seven segment display, DAC, keypad, elevator.
3. **Implement** Real Mode programming in Pentium.

****

ENHANCING SELF COMPETENCE

Sub Code : 16HU411	**Credits : 02**
Hrs/Week : 1+2+0+0	**Total Hours : 26**

Course Learning Objectives:

This Course will enable students to

1. Introspect and learn more about oneself
2. Learn social behaviour and etiquette
3. Develop positive attitude and values in life
4. Learn to be effective in communication and interactive skills
5. Educate on writing and presentation skills and also to educate onself on legal and ethical aspects

UNIT – I

Self Awareness and Emotional Quotient:
SWOT Analysis; Johari Window

4 Hours

UNIT – II

Grooming and Etiquette:
Personal grooming, hygiene, dressing for different occasions, making small talk, showing respect to women, eye contact, being appreciative, dos and don'ts in a conversation; Time Management.

4 Hours

UNIT – III

Attitude Development:
Building self worth, confidence, developing empathy; Goal Setting; Motivation.

5 Hours

UNIT – IV

Interactive Behavior:
Active listening, verbal & non-verbal communication, interview skills, group discussions, dealing with people in an organization, handling feed back and criticism.

7 Hours

UNIT - V

Writing and Presentation:
Formal and informal e-mails, framing requests, accepting or rejecting proposals, greetings, salutations, Close. Plagiarism, Presentation Skills.

6 Hours

Course Outcomes:

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

1. Develop awareness of his or her strengths and weaknesses and handle emotions.
2. Ensure a refined behaviour.
3. Become an asset to the society.
4. Become a good communicator.
5. Present to a group, on a one to one basis and create an impact.

REFERENCE BOOKS :

1. "Communicating at work – Principles and Practices for Business and the Professions" - Ronald B Adler & Jeanne Marquardt Elmhorst; McGraw-Hill College; Sixth Edition.
2. "Organizational Behaviour", - Stephen P Robbins; Prentice Hall, India.
3. "Organizational Behaviour", - Fred Luthans; McGraw Hill International Edition.

****