Kolhapur Institute of Technology's

College of Engineering (AUTONOMOUS), Kolhapur.

(An Autonomous Institute)

Syllabus

S. Y. B. Tech.

Computer Science & Engineering

SEM-I

Title of the Course: Computational Mathematics	L	T	P	Credits
Course Code: UCSE301	3	1	---	4

Course Pre-Requisite: Basics of Matrix Algebra, Rules and Formulae of Derivative, Basic Statistical Concepts, Set Theory.

Course Description: This Course contains Advanced Linear Algebra, Numerical Methods, Probability Distributions, Statistical Techniques and Fuzzy Sets.

Course Objectives:

1. To learn mathematical methodologies and models since mathematics is the foundation of engineering and technology.
2. To develop mathematical skills and enhance logical thinking power of students.
3. To provide students with skills in advanced linear algebra, probability, statistical techniques and fuzzy sets which would enable them to devise engineering solutions for given situations they may encounter in their profession.
4. To increase interest towards the use of mathematics in engineering module.

Course Outcomes:

COs	After the completion of the course the student will be able to
CO1	**Find** numerical solution of algebraic and transcendental equations using numerical method.
CO2	**Explain** the fuzzy sets and fuzzy logic in dealing with real problems.
CO3	**Make use of** method of least squares to fit the curves for given bivariate data.
CO4	**Apply** the knowledge of probability distributions to **solve** problems arising in engineering.
CO5	**Solve** the systems of simultaneous linear equations using factorization method.
CO6	**Determine** fuzzy numbers and **use** it in fuzzy equations.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2
CO1		2												
CO2	3													
CO3			3											
CO4		3												1
CO5				2										1
CO6			2											

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10

MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.
MSE: Assessment is based on 50% of course content (Normally first three units)
ESE: Assessment is based on 100% course content with60-70% weightage for course content (normally last three units) covered after MSE.

Course Contents:

Unit 1:--- Advanced Linear algebra 1. Revision of linear dependence of vectors. 2. Solutions of simultaneous linear equations using Gauss-Jordan method. 3. Solutions of simultaneous linear equations using LU decomposition method. 4. Determination of Eigen Value by Iteration method. 5. Solution of non-linear simultaneous equations.	7 **Hrs.**
Unit 2:--- Numerical methods 1. Solutions of algebraic and transcendental equations methods. 2. Bisection method. 3. Newton-Raphson method. 4. Secant method. 5. Numerical Integration. 6. Simpsons 1/3 and 3/8 rules. 7. Weddle's rule.	6 **Hrs.**
Unit 3:--- Probability and Distributions 1. Introduction of probability. 2. Laws of probability. 3. Conditional probability. 4. Baye's Theorem. 5. Random variables. 6. Discrete distributions: Binomial and Poisson. 7. Continuous distributions: Normal.	8 **Hrs.**

Unit 4:--- Statistical Techniques 1. Lines of regression of bivariate data, Correlation coefficient. 2. Fitting of Curves by method of Least-squares. 3. Fitting of Straight lines. 4. Fitting of Parabola. 5. Fitting of Exponential curves. 6. Tests of significations: Z-test, t-test (For single mean) 7. Chi-square test for independences of Attributes.	8 **Hrs.**
Unit 5:--- Introduction to Fuzzy sets and Fuzzy Logic 1. Crisp Sets: An overview. 2. Fuzzy sets: Basic concepts 3. Operations on fuzzy sets. 5. Multivalued Logics. 6. Inference from conditional fuzzy propositions.	7 **Hrs.**
Unit 6:--- Fuzzy Arithmetic 1. Fuzzy numbers. 2. Fuzzy cardinality 3. Operations on Fuzzy numbers. 4. Fuzzy equations of type A + X = B and A.X = B.	8 **Hrs.**
Reference Books: 1. Higher Engineering Mathematics by Dr. B. S. Grewal. 2. Linear Algebra by Seymour Lipschutz. 3. Fuzzy sets and Fuzzy Logic by George J. Klir, Bo Yuan. 4. Probability and Statistics for Computer science by James L. Johnon. 5. Fundamentals of Mathematical Statistics by Gupta and Kapoor.	
Unit wise Measurable Learning Outcomes:	

Unit 1:--- Advanced Linear algebra

Students are able to

a) Solve simultaneous linear and non linear equations.

b) Determine Eigen Value by Iteration method.

Unit 2:--- Numerical methods.

Students are able to

a) Evaluate integration numerically by Simpsons formulae.

b) Solve transcendental and algebraic equations by using numerical method.

Unit 3:--- Probability and Distributions.

Students are able to

a) Define random variable.

b) Verify the function as probability function.

c) useful to determine a reasonable distributional model for the data.

Unit 4:--- Statistical Techniques

Students are able to

a) Measure the correlation between bivariate data.

b) Apply fitting of curves for bivariate data.

c) Make use of Testing of Hypothesis.

Unit 5:--- Introduction to Fuzzy sets and Fuzzy Logic

Students are able to

a) Understand Basic concept of Fuzzy set theory.

b) Define membership functions.

c) Apply Basic operations on Fuzzy set.

Unit 6:--- Fuzzy Arithmetic

Student are able to

a) Determine Fuzzy numbers and Fuzzy cardinality.

b) Apply operate arithmetic operations on fuzzy numbers.

c) Solve fuzzy equations.

Title of the Course: Discrete Mathematical Structures	L	T	P	Credits

Course Code: UCSE302	3	1	---	4

Course Pre-Requisite: Mathematics - Probability theory, Set theory.

Course Description: This Course consists of concepts of Discrete mathematical structures such as Set theory, Algebraic systems, Lattices, Graphs, Counting theory etc.

Course Objectives:

1. To use mathematically correct terminology and notations.
2. To understand and critically analyze, formulate and solve the mathematical problems and proofs
3. To understand the concepts of Discrete Mathematics such as Sets, Algebraic Systems, Graphs, Groups and lattices
4. To design and implement experiments on Discrete Structures – Truth tables of statement formula, Set Operations, tree traversal techniques etc

Course Outcomes:

COs	After the completion of the course the student will be able to
CO1	Explain the discrete mathematical structures such as Sets, Algebraic systems, Groups, Probability in the field of Computer Science
CO2	Solve the problems related to the topics on discrete mathematics
CO3	Make use of discrete mathematical terminology and concepts in different areas of Computer Science.
CO4	Develop the implementation of functions and algorithms related to Discrete structures.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 01	PSO 02
CO1	3													
CO2		3												1
CO3		3											1	
CO4			2											1

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three units)

ESE: Assessment is based on 100% course content with60-70% weightage for course content (normally last three units) covered after MSE.

Course Contents:	
Unit 1: Mathematical logic (Text book-1) 1.1 Statements and Notations 1.2 Connectives – negation, Conjunction, disjunction, conditional, bi-conditional, Statement formulas and truth tables, well formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives 1.3 Normal and principal normal forms, completely parenthesized infix and polish notations 1.4 Theory of Inference for statement calculus – validity using truth table, rules of inference, consistency of Premises and indirect method of proof.	8 **Hrs**
Unit 2: Set theory (Text book-1) 2.1 Basic concepts of set theory, Operations on sets, Ordered pairs, Cartesian Products 2.2 Representation of discrete structures 2.3 Relation and ordering - properties of binary relations in a set, Relation matrix and the graph of a relation, Partition and Covering of set, Equivalence relations, Composition of Binary relations, Partial ordering , POSET and Hasse diagram. 2.4 Functions – types, composition of functions, Inverse functions.	8 **Hrs**
Unit 3: Algebraic systems (Text book-1) 3.1 Algebraic systems, properties and examples 3.2 Semigroups and Monoids, properties and examples, Homomorphism of Semigroups and Monoids 3.3 Groups: Definition and examples, Subgroups and homomorphism	5 **Hrs**
Unit 4: Lattices and Boolean algebra (Text book-1) 4.1 Lattice as POSETs , definition , examples and properties 4.2 Lattice as algebraic systems, Special lattices 4.3 Boolean algebra definition and examples 4.4 Boolean functions	5 **Hrs**
Unit 5: Permutations, Combinations and Probability theory (Text book-2) 5.1 The Basics of Counting 5.2 The Pigeonhole Principle 5.3 Permutations and Combinations 5.4 Generalized Permutations and Combinations 5.5 Discrete Probability	7 **Hrs**

5.6 Conditional probability 5.7 Bayes' Theorem	
Unit 6: Graphs (Text book-2) 5.1 Introduction to Graphs 5.2 Graph Terminology 5.3 Representing Graphs and Graph Isomorphism 5.4 Connectivity 5.5 Euler and Hamilton Paths 5.6 Planar Graphs 5.7 Introduction to Trees	7 **Hrs.**

Text Books:

1. Discrete Mathematical Structures with Application to Computer Science - J. P. Tremblay & R. Manohar (MGH International)
2. Discrete Mathematics and its Applications - Kenneth H. Rosen (AT&T Bell Labs) (mhhe.com/rosen)

Reference Books:

1. Discrete Mathematics - Semyour Lipschutz, MarcLipson (MGH), Schaum's outlines.
2. C. L. Liu and D. P. Mohapatra, "Elements of Discrete Mathematics", SiE Edition, TataMcGrawHill, 2008,ISBN 10:0-07-066913-9
3. Schaums Solved Problem Series – Lipschutz.
4. Discrete Mathematical Structures – Bernard Kolman, Robert Busby, S.C.Ross and NadeemurRehman (Pearson Education)

Title of the Course: Data Structures	L	P	T	Credits
Course Code: UCSE0303	3	-	-	3

Course Pre-requisite: Computer Programming

Course Objectives:

1. To learn basic concepts of C language.
2. To become familiar with advanced data structures such as Stacks, Queues, Trees etc.
3. To analyze and solve problems using advanced data structures such as Lists, Linked Lists, Queues, Stacks, Trees, and Graphs.
4. To write programs on Linked Lists, Doubly Linked Lists, Trees etc.

Course Outcomes:

CO	After completion of the course a student should be able to:
CO 1	Define the basic terms of Linear Lists, Linked List, Doubly Linked List, Non Linear Data Structures(Binary Trees, AVL Trees, Graphs)
CO 2	Choose the appropriate and optimal data structure for a specified application
CO 3	Analyze Time Complexity and Memory Complexity of different algorithms
CO 4	Write programs and applications with Static and Dynamic data structures

Mapping of course outcomes with program outcomes:

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO12	PSO1	PSO2
CO1	3	-	-	-	-	-	-	-	-	-	-	-	2	2
CO2	3	-	-	-	-	-	-	-	-	-	-	-	2	2
CO3	-	2	-	-	-	-	-	-	-	-	-	-	1	1
CO4	-	-	2	-	-	-	-	-	-	-	-	-	3	3

Assessment:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks

ISE-I	10
ISE-II	10
MSE	30
ESE	50

Course Contents:

Unit 1: Basics of Data Structure: Abstract Data Type (ADT), control structure, array, function, structure, pointer, Algorithm, Space and Time Complexity, Recursion, Towers of Hanoi and Ackermann's function, etc.	06 Hrs
Unit 2: Stacks and Queues: Stack: Definition, representation, implementation, applications of stack for expression evaluation and conversion Queue: Definition, representation, implementation, applications of queue, circular queue and priority queue	07 Hrs
Unit 3: Linked Lists: Definition, representation, implementation and operations on singly, doubly and circular linked lists, stack and queue implementation using linked list Hashing: Hashing functions, overflow handling, open and closed hashing, rehashing	09 Hrs
Unit 4: Searching and Sorting Techniques: Search: Importance of searching, types- sequential search, binary search Sort: Different types: bubble sort, selection sort, insertion sort, merge sort, quick sort, radix sort, heap sort	06 Hrs
Unit 5: Trees: Basic terminology, binary tree and its representation, binary tree traversal methods, binary search tree, AVL tree, B tree, B+ tree, Heaps and its operations.	07 Hrs
Unit 6: Graphs: Basic terminology and representation of graphs using adjacency matrix, storage representation, graph traversal techniques- Breadth First, Depth First	05 Hrs

Textbooks:

1. Data Structure using C- A. M. Tanenbaum, Y. Langsam, M. J. Augenstein (PHI)
2. Data Structures- A Pseudo code Approach with C – Richard F. Gilberg and Behrouz A. Forouzon, Cengage Learning, Second Edition.
3. Schaum's Outlines Data Structures – Seymour Lipschutz (MGH), Tata McGraw-Hill.

Reference books:

1. Fundamentals of Data Structures – Horowitz, Sahni CBS India
2. An introduction to data structures with Applications- Jean-Paul Tremblay, Paul. G. Soresan, Tata Mc-Graw Hill International Editions, Second Edition.

Title of the Course: Digital Logic Design & Microprocessors **Course Code: UCSE0304**	**L**	**T**	**P**	**Credit**
	4	-	-	4

Course Prerequisite: Fundamentals of Electronics and Computers, Basic Number System

Course Description:

The course is designed to provide knowledge of basic arithmetic and logical operations in digital systems, different sequential and combinational logic design. The subject provides fundamentals of 8085 & 80x86 Family Microprocessors.The subject gives idea of how assembly language programming works. This course is prerequisite for hardware based courses like Computer Architecture & Organization.

Course Objectives:

1) To provide knowledge of basic arithmetic and logical operations in digital systems.
2) To provide hands on knowledge about different sequential and combinational logicdesign.
3) To provide knowledge about construction & working of different microprocessors and peripheral.
4) To provide knowledge about assembly language programming.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Describe working of basic digital components
CO2	Illustrate different microprocessor operational & addressing modes
CO3	Analyze changes in microprocessor evolution
CO4	Develop Assembly Language Programs

CO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	3												1	
CO2			2											
CO3		2												
CO4		2												

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one EndSemester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on quiz / test.

MSE: Assessment is based on 50% of course content (Normally first three modules) ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.	
Course Contents:	
Unit 1:- Number Systems & Boolean Algebra Analog and digital systems, representation of signed numbers, 2's complement arithmetic, BCD addition & subtraction, octal & Hexadecimal addition and subtraction, Derived gates. Reduction of Boolean expressions, Boolean function representation, expansion of Boolean expression (standard SOP & POS),simplification of boolean expressions using K-map (upto 5 variable), Adders &Subtractors design using gates,	**8 Hrs.**
Unit 2:- Combinational & Sequential Logic Design Multiplexer, implementation of expression using MUX, Demultiplexer, decoder(74138), BCD to 7 segment decoder. Classification, Flip-flops(S-R, J-K,T,D)using gates, Race around condition Master –Slave J-K Flip Flop, Counters (Asynchronous & Synchronous), Design examples, Shift registers , State transition diagram, excitation table.	**8 Hrs.**
Unit 3:- 8085 Microprocessor Architecture The 8085 MPU, Microprocessor communication and bus timing, Demultiplexing address and Data bus, Generating control signals, The 8085 Architecture, opcode fetch machine cycle, memory read and write machine cycle. 8085 instruction groups, addressing modes.	**9 Hrs.**
Unit 4:- 8085 Programming Techniques Writing and execution assembly language program, counters & delays, Stack, Instruction related to stack execution of CALL and RET, The 8085 interrupt, RST instructions, vectored interrupts, RIM and SIM instructions . Basic interfacing concepts, peripherals I/O instructions IN, OUT, I/O execution, Memory - structure, interfacing & address decoding. Memory mapped I/O, I/O mapped I/O. The 8255 programmable peripheral interface, operating modes (I/O, BSR).	**10 Hrs.**
Unit 5:- 8086 Microprocessor and Assembly Language Architecture of 8086, Registers of 8086, Memory Model, Addressing Modes, Instruction Set, Programming Model	**7 Hrs.**
Unit 6:- 80x86 Family and Pentium Microprocessors The 80386 Microprocessor: The memory System, Special 80386 Registers Virtual 8086 Mode, The Memory Paging Mechanism, The Pentium Microprocessor : The Memory System, Special Pentium Registers, Pentium Memory Management, The Pentium Pro Microprocessor : Internal structure of the Pentium Pro The Pentium 4 : Memory Interface, Register Set, Hyper Threading Technology CPUID	**10 Hrs.**

Textbooks:

1. Fundamental of Digital Circuits –A. Anand Kumar, 2 nd Edition, PHI Private Limited.
2. Microprocessor architecture, programming & applications Ramesh S. Gaonkar, New Age International publication.
3. Microprocessors & Interfacing: Programming & Hardware, Douglas V. Hall, Tata McGraw Hill

References:

1. Digital fundamentals Floyd & Jain, , Pearson education, eighth edition, 2007
2. Digital Design –Morris Mano, Pearson Education
3. Modern Digital Electronics, R.P.Jain, 3rd Edition, Tata McGraw Hill, 2003
4. Digital systems, principles and applications – Ronald Tocci, Neal S. Widmer, Gregory Moss (Pearson Education) 9th Edition.

Unit wise Measurable students Learning Outcomes:
After learning these Unit **s**tudents will be able to-

1.1 Perform number system conversions and arithmetic operations in different number systems.
1.2 Analyze, expand or minimize boolean expression.
1.3 Use K-Map to simplify boolean expression.
2. Model different Flip-Flops and Counters.
3. Explain architectural details of 8085 microprocessor.
4.1 Classify Instruction set based on their purpose and size.
4.2 Determine addressing mode of 8085 Instructions.
4.3 Demonstrate programming skill in the assembly language programming using 8085 instruction set.
5.1 Explain architectural details of 8086 microprocessor.
5.2 Demonstrate programming skill in the assembly language programming using 8086 instruction set.
6. Analyze changes in different 80x86 family Microprocessors & Pentium Microprocessors.

Title of the Course: Data Communication and Networks Course Code: UCSE0305	L	T	P	Credit
	3	-	-	3

Course Pre-Requisite:

Course Description:

Course Objectives:

1. Help students understand basic components and devices of data communication system a
2. Study the layers in OSI and TCP/IP reference model
3. Study and implement the protocols and algorithms working at different layers in OSI and TCP/IP reference models

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain the basic concepts and components of data communication system
CO2	Compare and contrast various multiplexing and spreading techniques and transmission media at physical layer
CO3	Analyze various error detection and correction techniques at data link layer
CO4	Classify different multiple access protocols at medium access control sublayer

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3												1	
CO2	2	3			1									
CO3	2	3			1									
CO4	2	3			1								1	

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on Online objective test and quiz.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1:--- Introduction 1.1 Uses of Computer Networks 1.2 Network Hardware 1.3 Network Software	**6 Hrs.**

1.4 Reference Models	
Unit 2:- Communication Basics 2.1 Data & Signals :- Analog & Digital, Periodic analog signals, digital signals, Transmission Impairments, Data rate limits & Performance 2.2 Digital Transmission :- Line coding & line coding schemes (Unipolar, polar & bipolar) Transmission modes	**8 Hrs.**
Unit3 :- Multiplexing and Spreading 3.1 Multiplexing: Frequency-Division multiplexing, Wavelength-Division multiplexing, Synchronous Time-Division multiplexing, Statistical Time-Division multiplexing 3.2 Spread Spectrum: Frequency Hopping Spread spectrum(FHSS), Direct Sequence Spread Spectrum	**7 Hrs.**
Unit 4:--- Transmission Media 4.1 Transmission media :- Guided, Unguided media 4.2 Network Hardware components:- Transceivers & media converters, Repeaters, NIC & PC cards, Bridges, switches, Routers	**7 Hrs.**
Unit 5:--- Data Link Control Layer 4.1 Error Detection and Correction 4.2 Block Coding, Linear Block Codes 4.3 Cyclic Codes 4.4 Checksum 4.5 Data Link Control: Framing 4.6 Flow and Error Control 4.7 Protocols: Noiseless channels, Noisy Channels	**8 Hrs.**
Unit 6:-- The Medium Access Control Sublayer 5.1 Channel allocation Problem 5.2 Multiple Access Protocols: ALHOA, CSMA 5.3 Collision free protocols 5.4 Limited contention protocols.	**6 Hrs**
Textbooks: 1. Data Communications and Networking – Behrouz A Forouzan (The McGraw Hill) (Unit 2,3,4,5) 2. Computer Networks – Andrew S. Tanenbaum- (Prentice Hall) 5th Edition (Unit 1, 6)	
References: 1. Computer communications and Networking Technologies – Michael A Gallo (Cengage Learning) 2. Data & computer communications:- William Stallings (Pearson Education). 3. Data communication and computer Networks Ajit Pal (PHI Learning) .	

Title of the Course: Digital Logic Design & Microprocessors Lab Course Code: UCSE0331	L	T	P	Credit
	0	0	2	1

Course Prerequisite: Digital Logic Design & Microprocessors

Course Description: This subject covers practical details of subject Digital Logic Design & Microprocessors.

Course Objectives:

1. To provide hands on experience on construction of basic digital logic circuits
2. To provide knowledge about assembly language programming using 8085 instruction set.
3. To provide knowledge about assembly language programming using 8086 instruction set.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Model basic digital circuits
CO2	Develop simple assembly language programs using 8085 instruction set
CO3	Develop simple assembly language programs using 8086 instruction set

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	2												1	
CO2		2												
CO3		2												

Assessments :

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	50

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc.

ESE: Assessment is based on oral examination

Course Contents:

Experiment No. 1: Study of Basic & Universal Gates **Aim and Objectives:** i) Understand working of basic gates (OR, AND, NOT) ii) Understand working of universal gates (NOR, NAND) iii) Construct any gate using universal gates.	**2 Hrs.**

Outcomes: Students will be able to construct truth-table for basic / universal gates **Theoretical Background:** Logic Gates Characteristics **Experimentation:** i) Test different binary inputs to basic gates and observe the behaviour from output. ii) Construct basic gate using universal gates **Results and Discussions:** i)Truth Table for basic logic gates. ii) Truth-Table for universal gates iii) Compare Truth-Table of constructed circuit. **Conclusion:**i)We can verify characteristics of basic logic gates ii) We can construct any basic gate using universal gate	
Experiment No. 2:-Study of Boolean algebra & De Morgan's theorem **Aim and Objectives:** Understand Boolean algebra & De Morgan's theorem **Outcomes:** Students will be able to demonstrate De Morgan's theorem **Theoretical Background:** Boolean algebra & De Morgan's theorem **Experimentation:** Construct circuit to verify De Morgan's theorem using gates. **Results and Discussions:** Truth-Table for Boolean expression to verify De--Morgan's theorem. **Conclusion:** De Morgan's theorem is proved using Truth-Table	**2 Hrs.**
Experiment No. 3:-Study of R-S and J-K flip-flops **Aim and Objectives:** Construct R-S and J-K flip-flops **Outcomes:** Students will be able to implement R-S and J-K flip-flops **Theoretical Background:** Characteristics of different Flip-Flops **Experimentation:** Construct R-S and J-K flip-flops **Results and Discussions:** Truth Tables for R-S and J-K flip-flops **Conclusion:** Implemented circuit for R-S and J-K flip-flop	**2 Hrs.**
Experiment No.4:- Study of Counters **Aim and Objectives:** Implementing UP and DOWN counter **Outcomes:** Students will be able to implement UP and DOWN counter **Theoretical Background:** Characteristics and types of counter **Experimentation:** ConstructUP and DOWN counter **Results and Discussions:** Truth Tables for UP and DOWN counter **Conclusion:** Implemented circuit for UP and DOWN counter	**2 Hrs.**
Experiment No.5:-Interfacing counter circuitwith seven segment display **Aim and Objectives:** Interfacing counter circuit and seven segment display **Outcomes:** Students will be able to connect counter circuit to seven segment display **Theoretical Background:** Working of seven segment display **Experimentation:** Build interface for counter circuit and seven segment display **Results and Discussions:** Observation of output on seven segment display **Conclusion:** Built interface for counter circuit and seven segment display	**2 Hrs.**
Experiment No.6:-Realization of 4/5 variable K-maps **Aim and Objectives:** Minimizing 4/5 variable expression using K-Map **Outcomes:** Students will be able to minimize 4/5 variable expression. **Theoretical Background:** K-Map basics, Boolean algebra, De Morgan's theorem **Experimentation:** Use the K-map method minimize expression **Results and Discussions:** Verify minimized expression with Truth-Table **Conclusion:** Byfollowing K-map**,** expression can be minimized	**2 Hrs.**
Experiment No.7:- Study of 8085 microprocessor **Aim and Objectives:**Understand working of 8085 microprocessor	**2 Hrs.**

Outcomes: Students will be able to explain instructions of 8085 microprocessor **Theoretical Background:** Architecture & Instruction Set of 8085 microprocessor **Experimentation:** Use various instructions of 8085 microprocessor in simulator **Results and Discussions:** Table of Instructions with purpose, mnemonic & size **Conclusion:** Demonstrated instructions using simulator	
Experiment No.8:- 8085 Assembly Language Programs for - A. Addition / Subtraction of two 8-bit numbers B. Alter contents of register in 8085 C. Arrange no.s in ascending / descending order **Aim and Objectives:** Writing simple assembly language programs (listed above) **Outcomes:** Students will be able to develop simple assembly language programs **Theoretical Background:** Instruction Set of 8085 microprocessor **Experimentation:** Develop algorithm and program for given problem statements **Results and Discussions:** Execute developed programs and note the results **Conclusion:**	**2 Hrs.**
Experiment No.9:- 8085 Assembly Language Programs for - D. Block Transfer / Exchange E. Find square of numbers from 0 to 9 using table of a square F. Generate RST 7.5 interrupts **Aim and Objectives:** Writing simple assembly language programs (listed above) **Outcomes:** Students will be able to develop simple assembly language programs **Theoretical Background:** Instruction Set of 8085 microprocessor **Experimentation:** Develop algorithm and program for given problem statements **Results and Discussions:** Execute developed programs and note the results **Conclusion:**	**2 Hrs.**
Experiment No.10:- 8086 Assembly Language Programs for - A. Sorting an array for 8086 B. Searching for a number or character in a string for 8086 **Aim and Objectives:** Writing simple assembly language programs (listed above) **Outcomes:** Students will be able to develop simple assembly language programs **Theoretical Background:** Instruction Set of 8086 microprocessor **Experimentation:** Develop algorithm and program for given problem statements **Results and Discussions:** Execute developed programs and note the results **Conclusion:**	**2 Hrs**
Experiment No.11:- 8086 Assembly Language Programs for - A. String manipulations for 8086 B. Digital clock design using 8086 **Aim and Objectives:** Writing simple assembly language programs (listed above) **Outcomes:** Students will be able to develop simple assembly language programs **Theoretical Background:** Instruction Set of 8086 microprocessor **Experimentation:** Develop algorithm and program for given problem statements **Results and Discussions:** Execute developed programs and note the results **Conclusion:**	**2 Hrs**
Experiment No. 12: Interfacing 8086 with -	

A. ADC and DAC B. Stepper Motor **Aim and Objectives:** Writing simple assembly language programs (listed above) **Outcomes:** Students will be able to develop simple assembly language programs **Theoretical Background:** Instruction Set of 8086 microprocessor **Experimentation:** Develop algorithm and program for given problem statements **Results and Discussions:** Execute developed programs and note the results **Conclusion:**	**2 Hrs**

Textbooks:

1. Fundamental of Digital Circuits –A. Anand Kumar, 2 nd Edition, PHI Private Limited.
2. Microprocessor architecture, programming &applicationsRamesh S. Gaonkar, New Age International publication.
3. Microprocessors & Interfacing: Programming & Hardware, Douglas V. Hall, Tata

References:

1. Digital fundamentals Floyd & Jain, , Pearson education, eighth edition, 2007
2. Digital Design –Morris Mano, Pearson Education
3. Modern Digital Electronics, R.P.Jain, 3rd Edition, Tata McGrawHill, 2003
4. Digital systems, principles and applications – Ronald Tocci, Neal S. Widmer, Gregory Moss (Pearson Education) 9 th Edition.

Title of the Course: Data Communication and Networking Lab	L	T	P	Credit
Course Code: UCSE0332	-	-	2	1

Course Pre-Requisite:

Course Description: Study and implement basic techniques in data communication system

Course Objectives:

To expose students to:-

1. Basic components of data communication system
2. Networking devices and topologies
3. Layered structure of computer networks
4. Functionalities of Physical and Data Link Layer

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Design sample network based on organizations requirements
CO2	Demonstrate working of different interconnecting devices using simulation tools
CO3	Make use of different network testing tools and commands for sample network testing and analysis
CO4	Design program for framing, flow control and error correction and detection techniques using programming language

CO-PO Mapping:

CO	PO 1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1			2	1		1		1						
CO2					3			1					2	
CO3					2			1					2	
CO4			2					1						

Assessments :

Teacher Assessment:

One components of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Marks	Assessment
25	ISE
50	ESE(POE)

Course Contents:

Experiment No. 1:- Campus Network Design using CISCO Packet Tracer	**2 Hrs**

Experiment No. 2:- Demonstration of Interconnecting Devices using CISCO Packet Tracer	
Experiment No.3:- Study of connectivity test tools with all its option	**2 Hrs**
Experiment No. 4:- Wireshark Network Protocol Analyzer	**2 Hrs**
Experiment No. 5:- Design and Implementation of Framing Techniques A) Character Count B) Bit Stuffing	**2 Hrs**
Experiment No. 6:- Design and Implementation of Error Detection and Correction Codes A) Cyclic Redundancy Check B) Hamming Code	**2 Hrs**
Experiment No. 7:- Design and Implementing Elementary data link protocol (Stop & wait protocol)	**2 Hrs**
Experiment No. 8:- Design and Implementing Elementary data link protocol (Go Back N)	**2 Hrs**
Experiment No. 9:- Design and Implementing Elementary data link protocol (Selective Repeat)	**2 Hrs**

Title of the Course: Data Structures Lab	L	P	T	Credits
Course Code: UCSE0333	**-**	**4**	**-**	**2**

Course Pre-requisite: Computer Programming

Course Objectives:

1. To learn basic concepts of C language structures, Arrays, lists pointers.
2. To become familiar with advanced data structures such as Stacks, Queues, Trees etc.
3. To analyze and solve problems using advanced data structures such as Lists, Linked Lists, Queues, Stacks, Trees, and Graphs.
4. To write programs on Linked Lists, Doubly Linked Lists, Trees etc.

Course Outcomes:

CO	After completion of the course a student should be able to:
CO 1	Define the basic terms of Linear Lists, Linked List, Doubly Linked List, Non Linear Data Structures(Binary Trees, AVL Trees, Graphs)
CO 2	Choose the appropriate and optimal data structure for a specified application
CO 3	Write programs and applications with Static and Dynamic data structures

Mapping of course outcomes with program outcomes:

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO12	PSO1	PSO2
CO1	3	-	-	-	-	-	-	-	-	-	-	-	1	1
CO2	3	-	-	-	-	-	-	-	-	-	-	-	1	1
CO3	-	-	2	-	-	-	-	-	-	-	-	-	2	2

Assessment:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weight age respectively.

Assessment	Marks
ISE	50
ESE(POE)	50

Course Contents:

Assignments based on topics covered in course UCSE0303 Data Structures:

1. Program based on arrays, structures and pointers.
2. Program based on functions and recursion.
3. Program for developing an application using stack.
4. Program for developing an application using queue and circular queue.
5. Program for developing an application using singly linked list.
6. Program for developing an application using doubly linked list.
7. Program for developing an application using circularly linked list.
8. Program based on implementation of hashing and rehashing.
9. Program based on implementation of linear search, binary search.
10. Program based on one of the sorting techniques.
11. Program based on one of the sorting techniques.
12. Implementation of recursive and non-recursive tree traversals.
13. Implementation of basic binary search tree and its application.
14. Program based on AVL tree / B-tree.
15. Program based on representation of graphs
16. Program based on DFS and BFS search.

Textbooks:

1. Data Structures- A Pseudo code Approach with C – Richard F. Gilberg and Behrouz A. Forouzon, Cengage Learning, Second Edition.
2. Schaum's Outlines Data Structures – Seymour Lipschutz (MGH), Tata McGraw-Hill.
3. The C Programming langauge – Kernighan and Ritchie

Reference books:

1. Data Structure using C- A. M. Tanenbaum, Y. Langsam, M. J. Augenstein (PHI)
2. An introduction to data structures with Applications- Jean-Paul Tremblay, Paul. G. Soresan, Tata Mc-Graw Hill International Editions, Second Edition.

SEM-II

Title of the Course: Automata Theory	L	T	P	Credit
Course Code: UCSE0401	3	1		4

Course Pre-Requisite: Discrete Mathematics, Sets, Cartesian Product and Functions

Course Description: This course deals with the theoretical background of computer science.

Course Objectives:

1. To expose the students to the mathematical foundations and principles of computer science..
2. To strengthen the students' ability to carry out formal and higher studies in computer science
3. To make the students understand the use of automata theory in Compliers & System programming.
4. To make the student aware of mathematical tools, formal methods & automata techniques for computing.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain types of formal languages and their acceptors
CO2	Classify formal languages on the basis of their features
CO3	Relate the computational models with the modern day computer technologies
CO4	Design computational machines of various types for specified problems

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2													
CO2	2													
CO3		2												1
CO4			3											1

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one EndSemester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

UNIT-I : Mathematical Induction, Regular Languages & Finite Automata:	08 Hrs.
The Principle of Mathematical Induction Recursive Definitions, Definition & types of	

grammars & languages, Regular expressions and corresponding regular languages, examples and applications, unions, intersection & complements of regular languages, Finite automata-definition and representation, on-deterministic F.A.,NFA with null transitions, Equivalence of FA's , NFA's and NFA's with null transitions.	
UNIT-II: Kleene's Theorem: Part I & II statements and proofs, minimum state of FA for a regular language, minimizing number of states in Finite Automata.	**04 Hrs.**
UNIT-III: Grammars and Languages: Derivation and ambiguity, BNF & CNF notations, Union, Concatenation and *'s of CFLs, Eliminating production & unit productions from CFG, Eliminating useless variables from a context Free Grammar. Parsing: Top-Down, Recursive Descent and Bottom-Up Parsing	**10 Hrs.**
UNIT-IV: Push Down Automata Definition, Deterministic PDA & types of acceptance, Equivalence of CFG's & PDA's.	**04 Hrs.**
UNIT-V: CFL's and non CFL's : Pumping Lemma and examples, intersections and complements.	**04 Hrs.**
UNIT-VI: Turing Machines: Models of computation, definition of Turing Machine as Language acceptors, combining Turing Machines, Computing a function with a TM, Non-deterministic TM and Universal TM, Recursively enumerable languages, Unsolvable problems.	**10 Hrs.**

Textbooks:

1. Introduction to languages & Theory of computations – John C. Martin (MGH) –Chapters 1, 2,3,4,5,6,7,8.
2. Discrete Mathematical Structures with applications to Computer Science—J .P.Trembley & R.Manohar (MGH) Chapter 1,

References:

1.Introduction to Automata Theory , Languages and computation – John E. Hopcraft , Rajeev Motwani , Jeffrey D. Ullman (Pearson Edition).
2.Introduction to Theory of Computations – Michael Sipser (Thomson Brooks / Cole)
3.Theory Of Computation- Vivek Kulkarni, 1st edition OXFORD university Press
4.Theory Of Computation A problem Solving Approach Kavi Mahesh Wiley India

Title of the Course: Computer Graphics	L	T	P	Credit
Course Code: UCSE0402	3			3

Course Pre-Requisite:

Course Description: Study basic and core concepts in Computer Graphics

Course Objectives:

1. To provide students with an understanding of various transformation techniques and projections.
2. To provide students with an understanding of various algorithms related to drawing line, circle, polygon scanning, filling, windowing and clipping of graphical objects.
3. To provide students with an understanding of the mathematics underlying two- and three-dimensional interpolating curves and to learn the Warnock and depth-buffer (Z-buffer) algorithm used to determine hidden lines and surfaces in a rendered scene
4. To enable students to acquire practical knowledge in animation, illumination, lightening and rendering using OpenGL.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain the basic concepts of interactive computer graphics.
CO2	Illustrate the core concepts of computer graphics, including viewing, projection, perspective and transformation in two and three dimensions.
CO3	Apply the mathematical foundations to interpolate parametric and non-parametric curves and surfaces.
CO4	Analyze basic illumination models and polygon rendering methods.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO 12	PS O1	PS O2
CO1	3				2									
CO2	3	3	2	1									2	3
CO3	3	3	2	1									3	3
CO4		3	2	2									3	3

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one EndSemester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on Online objective test, presentation, seminar, quiz etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction	6 Hrs.
1.1 Overview of graphics systems – Video display devices, 1.2 Raster scan systems	

1.3 Random scan systems 1.4 Graphics monitors and Workstations, 1.5 Input devices, Hard copy Devices, Graphics Software	
Unit 2 : Transformations 2.1 Basic 2D & 3D transformations - Translation, Scaling, Rotation, Reflection, Shearing, Multiple Transformations 2.2 Rotation about an axis parallel to a coordinate axis 2.3 Rotation about an arbitrary axis in space 2.4 Affine and Perspective Geometry 2.5 Orthographic projections 2.6 Axonometric projections.	**8 Hrs.**
Unit 3: Raster Scan Graphics 3.1 Bresenhams Line drawing algorithm 3.2 Bresenhams Circle drawing algorithm 3.3 Scan Conversion techniques: RLE, Frame Buffer 3.4 Scan converting polygons: Edge fill and Seed fill algorithms 3.5 Anti-aliasing	**8 Hrs.**
Unit 4: Viewing and clipping 4.1 Introduction 4.2 Windowing and View-porting, 4.3 Sutherland - Cohen line clipping algorithm	**4 Hrs.**
Unit 5 : Curves and Surfaces 5.1 Non-parametric and parametric curves 5.2 Representation of space curves 5.3 Cubic Spline 5.4 Parabolic Blended curves 5.5 Bezier curves 5.6 B-spline curves 5.7 Z- buffer algorithm 5.8 Warnock algorithm	**8 Hrs.**
Unit 6 : Illumination models and surface rendering methods 6.1 Light sources 6.2 Basic illumination models 6.3 Displaying light intensities 6.4 Halftone patterns and Dithering Techniques 6.5 Polygon Rendering methods 6.6 Ray tracing methods	**8 Hrs.**
Textbooks: 1. Computer Graphics C Version second edition –Donald D. Hearn, M. Pauline Baker (Pearson) 2. Mathematical elements for Computer Graphics - David F. Rogers, J. Alan Adams (MGH	

International)
3. Procedural elements for Computer Graphics - David F. Rogers (MGH International)

References:

1. Principles of Computer Graphics Theory and Practice Using OpenGL and Maya, Shalini Govil-Pai, (Springer) .
2. Computer Graphics (second Edition) - Zhigang Xiang & Roy Plastock (Schaum's Outline Series, TMGH).
3. Computer Graphics Using OpenGL F.S. Hill Jr. Stephen M. Kelley, (Pearson Education).

Unit Wise Measurable Students Learning Outcomes:

1 Explain the graphics devices
2 Explain 2D and 3D transformations
3 Explain algorithms for drawing line, circle and polygon filling
4 Explain algorithms for clipping and hidden line elimination
5 Explain mathematical representation of plane curves and space curves
6 Explain different illumination models

Title of the Course: Computer Networks Course Code: UCSE0403	L	T	P	Credit
	3	-	-	3

Course Pre-Requisite:

UCSE0305 Data Communication and Networks

Course Description: This course provides a solid understanding of each of the most important networking protocols within the IP suite. The Internet protocol suite provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed and received.

Course Objectives:

1: To make students able to identify client-server model and implement it using socket programming.

2: To introduce students with emerging protocols IPv6 and the ICMPv6 and write applications to communicate using IPv6.

3: To make students familiar with architecture and working of protocols like IP, TCP, UDP, DHCP, DNS, FTP, WWW

4: To make students able to understand working of email system and write an application to send and receive e-mail

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Recall the basic concept of Network, Transport and Application Layer.
CO2	Describe different terminologies of client server programming
CO3	Illustrate different application layer protocol like DHCP, DNS, FTP, HTTP, SMTP and SNMP.
CO4	Describe various protocols supported by multimedia content.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1													
CO2			2										2	
CO3	3												1	
CO4	2													

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and **ISE 2** are based on Online objective test and quiz.
MSE: Assessment is based on 50% of course content (Normally first three modules) **ESE**: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

UNIT-I : : Network Layer Introduction, Network layer services, addressing, IP packet format, ARP, RARP, ICMP, Packet routing protocols, congestion control, IPv6- Introduction, addressing, transition from IPv4 to IPv6	**8 Hrs.**
UNIT-II: Transport Layer Transport layer functions, UDP- datagram, services, applications, TCP - services, segment, connection, state transition diagram, Flow control, congestion control, error control, timers.	**8Hrs.**
UNIT-III: Introduction to Application Layer Client-Server paradigm, client, server, concurrency, socket interface, communication using TCP, communication using UDP	**6 Hrs.**
UNIT-IV: DHCP, DNS, FTP and TFTP DHCP: Introduction, Previous Protocols, DHCP operation, Packet Format, DHCP Configuration. DNS: Need, Name Space, Domain Name Space, Distribution of name space, and DNS in internet, Resolution, DNS massages, Types of records, Compression examples, encapsulation. FTP: Connections, Communication, Command processing, File transfer, User interface, Anonymous FTP, TFTP.	**9 Hrs.**
UNIT-V: HTTP, Electronic Mail, SNMP HTTP: Architecture, Web Documents, HTTP Transaction, Request & Response messages: header & examples, Persistent vs. non persistent HTTP, Proxy Servers. Architecture, User agents, addresses, delayed delivery, Aliases, Mail transfer agent SMTP commands & responses, mail transfer phases, MIME, Mail Delivery, mail access protocols, SNMP.	**9 Hrs.**
UNIT-VI: Multimedia in Internet: Streaming stored audio/video, streaming live audio/video, real-time interactive audio/video, real-time transport protocol (RTP), real-time transport control protocol (RTCP), voice over IP (VoIP): session initiation protocol (SIP) and H.323.	**8 Hrs**

Textbooks:

1. TCP/IP Protocol Suite by B. A. Forouzan, TMGH Publication

References:

1. Computer Networks by Andrew Tanenbaum, PHI Publication
2. Computer Networks by William Stallings, PHI Publication

Title of the Course: Computer Organization and Architecture Course Code:UCSE0404	L	T	P	Credit
	3	-	-	3

Course Pre-Requisite: Digital Logic Design (UCSE0304), Digital Logic Design Lab (UCSE0331)

Course Description:

This course will introduce students to the fundamental concepts of modern computer organization and architecture. Course introduces hardware design, basic structure and behavior of the various functional modules of the computer and how they interact to provide the processing needs of the user. It covers instruction sets, CPU structure and functions, memory system organization and architecture, multiprocessor systems. The emphasis is on studying and analyzing fundamental issues in architecture design and their impact on performance.

Course Objectives:

1. To expose students to basic concepts of computer organization.
2. To provide a comprehensive and self contained view of control unit design.
3. To analyze performance issues in processor and memory design of a digital computer.
4. To learn concepts of pipeline architectures and different performance measures.
5. To understand parallel and distributed memory architectures.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain the organization of basic computer and its function, instruction types and data formats
CO2	Design a simple control unit for the given task by applying the theory concepts.
CO3	Analyze some of the design issues in terms of speed, technology, cost, performance.
CO4	Illustrate memory organization and memory management hardware.
CO5	Learn the concepts of parallel, pipelined and distributed computer architectures.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	2													
CO2	2		2											
CO3	2													
CO4		3		3										
CO5		2												

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10
MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on assignment/declared test/quiz/seminar/Group Discussions etc. MSE: Assessment is based on 50% of course content (Normally first three modules) ESE: Assessment is based on 100% course content with60-70% weightage for course content (normally last three modules) covered after MSE.	
Course Contents:	
Unit 1: Basic Computer Organization Evolution of computers - Electronic computers-generations, VLSI era , CPU organization , user and supervisor modes, accumulator based CPU, System bus, types of instruction(zero, one, two and three address machines), RISC& CISC, definition, comparison and examples, Data representation: Fixed - Point Numbers, Floating Point Number- The IEEE 754 floating pointing numbers	**7 Hrs.**
Unit 2: Hardwired Control Design Introduction, multi cycle operation, implementation methods, Hardwired control, design methods, state tables, GCD processor, Classical method, one hot method, Design example twos complement multiplier control, CPU control unit design	**6 Hrs.**
Unit 3: Microprogrammed Control Design Basic concepts, control unit organization, parallelism in microinstructions, Microinstruction addressing, timing, Control unit organization, Design example-twos complement, multiplier control, Control field encoding, encoding by function, multiple microinstruction formats.	**6 Hrs.**
Unit 4: Memory Organization Types of memory, Memory systems, multi level, address translation, memory allocation, Caches, Associative memory, direct mapping, set associative addressing	**9 Hrs.**
Unit 5: Introduction to Pipeline and Parallel Processing Pipelining, linear pipelining, classification of pipeline processors Interleaved memory organizations, performance evaluation factors. Parallel Processors-Flynn's Classification, Introduction to Associative memory processors,	**6 Hrs.**
Unit 6:Distributed Memory Architecture Loosely coupled and tightly coupled architectures. Cluster computing as an application of loosely coupled architecture. Examples – CM*.	**7 Hrs.**
Textbooks: 1. Computer Architecture and Organization - John P Hayes (MGH) 3rd Edition 2. Advanced computer architecture – Kai Hwang(MGH)	
References: 1] Computer Architecture & Parallel Processing – Kai Hwang & Briggs (MGH) 2] Computer Organization - Hamacher Zaky (MGH).	
Unit wise Measurable students Learning Outcomes: 1 Students will be able to understand the basic organization of computer system. 2 Students will be able to differentiate between control unit design methods. 3 Students will be able to describe function of microprogrammed control unit. 4 Students will be able to differentiate and evaluate performance of various memory levels. 5 Students will be able to explain various parallel processing architectures. 6 Students will be able to understand the distributed memory architectures.	

Title of the Course: Software Engineering Course Code: UCSE0405	L	T	P	Credit
	3	-	-	3

Course Pre-Requisite:

Course Description: This course provides basic concepts, principles of software engineering & basics of Project Management.

Course Objectives:

1. To expose the students to basic concepts, principles of software engineering & importance of SDLC in their project development work.
2. To expose the students to software testing techniques and software quality management.
3. To introduce students basics of Object Oriented Modeling and Design.
4. To make the student aware of role of Software Engineering in Project Management.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Explain the Software Development Process.
CO2	Illustrate the Software Testing techniques and Quality Assurance in detail
CO3	Make use of Project management Concepts in the project development.
CO4	Design the solution to the problems using Object Oriented Modelling with UML.

Mapping of Course Outcomes with Program Outcomes and Program Specific Outcomes

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	3											-		
CO 2												-		
CO 3									2		3	-	1	
CO 4		2	3	1	1				3		2		1	2

Assessments :

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE 1	10

MSE	30
ISE 2	10
ESE	50

ISE 1 and ISE 2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: The software Problem [Text Book-1] 1.1 Software Problems, Software Engineering Problems 1.2 Cost, schedule & Quality, Scale and change, 1.3 Software Development process Modules, 1.4 Project Management Process 1.5 Software Processes: Process & Project	**5 Hrs.**
Unit 2: Requirements Analysis & specification [Text Book-1,2] 2.1 Requirements gathering & Analysis 2.2 Software Requirements Specifications 2.3 Collecting Requirements, Defining Scope, 2.4 Creating the Work Breakdown Structure, Validating Scope, Controlling Scope 2.5 Basic Principles of Cost Management, 2.6 Planning Cost Management, Estimating Costs, 2.7 Determining the Budget, Controlling Costs, Formal System Development Techniques	**6 Hrs.**
Unit 3: Design [Text Book-1,2] 3.1 Design Concepts 3.2 Function Oriented Design 3.3 Object Oriented Design 3.4 Detail Design 3.5 Verification 3.6 Metrics	**7 Hrs.**
Unit 4: Object Oriented Modeling and Design [Text Book-3] 4.1 Object Oriented Design : What is Object Orientation? What is OO Development? OO Themes. 4.2 Modeling as Design Techniques: Modeling, Abstraction, Three Models 4.3 Overview of UML **[Text Book-4]** 4.4 Conceptual Model of UML 4.5 Architecture	**6 Hrs.**
Unit 5: Coding & Testing [Text Book-1] 4.1 Coding & Code Review 4.2 Testing 4.3 Unit Testing 4.4 Black Box Testing	**7 Hrs.**

4.5 White Box Testing 4.6 Integration Testing 4.7 System Testing	
<u>Unit 6: Quality Management</u> [Text Book-2,1] 5.1 Importance, Planning Quality Management, 5.2 Performing Quality Assurance, Controlling Quality, 5.3 Tools and Techniques for Quality Control, 5.4 Modern Quality Management, Improving IT Project Quality 5.5 ISO 9000 SEI capability Maturity Model, Six Sigma 5.6 Agile software Development & Extreme Programming 5.7 Agile Project Management	**7 Hrs.**

Textbooks:

1. 1 Software Engineering : A precise Approach - Pankaj Jalote (Wiley India)
2. Information Technology Project Management, 7E, Kathy Schwalbe, Cengage Learning (India Edition)
3. Object Oriented Modeling and Design with UML, Michel R Blaha, James R Rambaugh, Second Edition
4. The Unified Modelling Language User Guide: - Grady Booch, James Rambaugh, Lvar Jacobson.

References:

1. IT Project Management, 3 E, Joseph Phillips, McGraw Hill Edu. (India) Pvt. Ltd.
2. Software Project Management, Bob Huges, Mike Cotterell, Rajib Mall, 5/E, Tata McGraw Hill Edu. (India) Pvt. Ltd.

Unit wise Measurable students Learning Outcomes:

After learning this unit the student will be able to-

1. Explain SDLC.
2. Explain software testing techniques and software quality management.
3. Illustrate basics of Object Oriented Modelling concepts.
4. Explain role of Quality in Project Management.

Title of the Course: Object Oriented Progg. Using C++ Lab. Course Code:UCSE0431	L	T	P	Cr
	2	0	2	3

Course Prerequisite:

- Knowledge of C programming

Course Description:

This course exposes students to the concepts of Object Oriented Programming (OOP). It helps students to choose proper OOP concepts to solve different problems. Upon completion, students should be able to write efficient, reusable programs for a given problem using OOP concepts.

Course Learning Objectives:

1. To expose the students to concepts of Object Oriented Paradigm.
2. To make students understand the use of programming constructs of C++.
3. To give hands on exposure to develop applications based on concepts of Object Oriented approach.

Course Outcomes:

CO	After the completion of the course the student should be able to
CO1	explain object oriented concepts, principles and techniques.
CO2	select appropriate approach from procedural and object oriented to solve the given problem.
CO3	apply various object oriented features to solve real life problems using C++ language.
CO4	make use of exception handling and STL to solve given problems.

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	1											1	1	
CO2	2	1										1		
CO3	1	2	3		1			1		1		1	1	3
CO4	1	2	2		2			1				1	1	2

Assessments :

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	50
ESE	50

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc.

ESE: Assessment is based on practical and oral examination

Course Contents:

Unit 1: Introduction: Introduction to procedural & object-oriented programming, Limitations of procedural programming, Need of object-oriented programming,	03 **Hrs.**

Fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism.	
Unit 2: Basics of C++ programming: Friend Functions, Friend Classes, Inline Functions, Parameterized constructors, Static class members, Scope resolution operators, Passing objects to functions, nested classes, and local classes.	04 **Hrs.**
Unit 3: Inheritance: Need of Inheritance, Concept, public, private, protected inheritance, Single inheritance, Multiple and multilevel inheritance, Hybrid Inheritance, Virtual base class, overriding of member functions, static variable, static function, friend function, friend class	04 **Hrs.**
Unit 4: Polymorphism: Pointers basics of memory management, New and delete operators, Pointer to object, Pointer to data members, this pointer. Need of Polymorphism, concept, Compile time polymorphism or early binding: function overloading and operator overloading, operator overloading using member function and friend function, overloading - unary, binary, arithmetic operators, relational operators, Overloading new and delete operators, insertion and extraction operators, Run time polymorphism or late binding using Virtual function, pure virtual function, Abstract class, Type conversion	06 **Hrs.**
Unit 5: Files and Streams: Concept of Streams, concept of File, opening and closing a file, detecting end-of-file, file modes, file pointer, reading and writing characters, strings and objects to the file, operations to move file pointers i.e seekg, seekp, tellg, tellp.	05 **Hrs.**
Unit 6: Advanced C++ features: Introduction to Generic Programming using Templates: Function template and class template, Introduction to Standard Template Library (STL), containers, iterators and algorithms, study of container template classes for vectors and stacks and related algorithms 2 Exception handling: Introduction, syntax for exception handling code: try-catch-throw, Multiple Exceptions, Exceptions with arguments	06 **Hrs.**

Textbooks:

1. C++ programming by Robert Lafore 4th Edition (SAMS)
2. The Complete Reference: C++ - Herbert Schildt (TMGH) Fourth Edition.

References:

1. C++ Programming with language - Bjarne Stroustrup, AT & T
2. Object oriented Programming in C++ 3rd Edition-R.Lafore (Galgotia Publications)
3. C++programming –John Thomas Berry(PHI) • Object –Oriented Analysis & Design: Understanding System Development with UML 2.0 , Docherty, Wiley India Ltd.
4. http://www.spoken-tutorial.org/ NMEICT Project of Govt. Of India.

Assignments & Laboratory Work:

Minimum 10-12 Experiments are to be performed in batches, on the above topics. Term work should comprise detailed documentation on the below 10-12 experiments. Students should implement programs based on the following topics preferably on Linux platform.

1. Study of OOP features and compare it with POP.
2. Functions with default (Optional) arguments.
3. Classes (with constructor) and Objects.
4. Operator Overloading.
5. Inheritance
6. Memory Management
7. Polymorphism
8. Type Conversion
9. Exception Handling
10. Template
11. File Handling
12. STL

Title of the Course: Computer Graphics Lab	L	T	P	Credit
Course Code: UCSE0432			2	1

Course Pre-Requisite: C programming and mathematics

Course Description: Study and implement basic and core techniques in Computer Graphics

Course Objectives: To expose students to:-

1. Understand the need of developing graphics application
2. Learn algorithmic development of graphics primitives like: line, circle, polygon etc.
3. Learn the representation and transformation of graphical images and pictures.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Develop effective OpenGL programs to solve graphics programming issues, including 3D transformation, objects modelling, colour modelling, lighting, textures, and ray tracing.
CO2	Make use of modern tools such as blender, adobe flash for developing computer graphics applications.
CO3	Illustrate effects of various illumination models and ray tracing methods in computer graphics.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	3	2	3								3	3
CO2	2	2	3	2	3								3	3
CO3	2	2	3	2	3								3	3

Assessments :

Teacher Assessment:

One components of In Semester Evaluation (ISE) and one EndSemester Examination (ESE) having 50% and 50% weights respectively.

Marks	Assessment
25	ISE
25	ESE(OE)

Course Contents:

Experiment No. 1:- : OpenGL programming to use basic graphics primitives	**2 Hrs**
Experiment No. 2:- Write a menu driven program in C to implement two dimensional transformation on two dimensional objects like rotation, reflection, scaling and shearing using Open GL	**2 Hrs**

Experiment No. 3:- Install the necessary packages in CentOS/Ubuntu for running the graphics program and include graphics.h header/libraries file in gcc compiler	**2 Hrs**
Experiment No.4:- Implementing Bresenham's line drawing algorithm.	**2 Hrs**
Experiment No. 5:- Implementing Bresenham's circle generation algorithm.	**2 Hrs**
Experiment No. 5:- Implementing Edge fill algorithm.	**2 Hrs**
Experiment No. 6:- Implementing Seed fill algorithm.	**2 Hrs**
Experiment No. 7:- Implementing Sutherland-Cohen line clipping algorithm.	**2 Hrs**
Experiment No. 8:- Implementing Basic Illumination Models	**2 Hrs**
Experiment No. 9:- Implementing Basic Ray Tracing algorithm.	**2 Hrs**
Textbooks: 1. Computer Graphics Using OpenGL F.S. Hill Jr. Stephen M. Kelley, (Pearson Education).	
Unit wise Measurable students Learning Outcomes:	

Title of the Course: Computer Networks Lab Course Code: UCSE0433	L	T	P	Credit
	-	-	2	1

Course Pre-Requisite: Data Communication and Networking Theory & Lab.

Course Description: Study top four layers of OSI networking model and implement example programs at different layers and use different networking tools.

Course Objectives: To expose students to:-

1. Basic concepts of Client Server model of Internet using Socket programming
2. Logical addressing of computers/nodes in LAN/WAN.
3. Application layer protocols such as HTTP, FTP, TELNET, DHCP etc.
4. Networking tools such as Packet Tracer TCPDUMP and Wireshark to analyze protocols.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Design network for an organization as per the requirements
CO2	Design UDP and TCP client server program to demonstrate simple ,iterative and concurrent server
CO3	Demonstrate working of different routing protocols and application layer protocols using Wireshark/Packet Tracer/TCPDump
CO4	Design client client server program to send and receive email, web pages.
CO5	Install and Configure FOSS server to provide different services

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2							1	2	2		1		2
CO2			3					1						
CO3		1			2			1					2	
CO4			3		2			1				1	2	
CO5					2	1	1	1						

Assessments :

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50% and 50% weights respectively.

Marks	Assessment
25	ISE
25	ESE(OE)

Course Contents:

Experiment No. 0:- Socket Programming API in C/C++	**2 Hrs**
Experiment No. 1:- Well Known Server and Client	**2 Hrs**
Experiment No. 2:- Routing Algorithm and Protocols **A)** Implementation of Shortest Path routing algorithm in C / C++ programming language **B)** Implementation of Distance Vector routing algorithm in C / C++ programming language. **C)** Simulation of Routing Protocols	**4 Hrs**
Experiment No. 3:- Implementation of C program to find Network ID, Host ID and the network Class of a given input IPv4 Address.	**2 Hrs**
Experiment No. 4:- Implementation of Iterative Client / Server Model using TCP Sockets	
Experiment No. 5:- Implementation of Concurrent Client / Server Model using TCP Sockets.	**2 Hrs**
Experiment No. 6:- Implementation of Client / Server Model using UDP sockets.	**2 Hrs**
Experiment No. 7:- Communication using IPv6	**2 Hrs**
Experiment No. 8:- Packet Capturing and Analysis	**2 Hrs**
Experiment No. 9:- Demonstration of working of DHCP,DNS,FTP,SSH,TELNET protocols	**2 Hrs**
Experiment No. 10 :- Install and Configure different types of services on FOSS Server	**2 Hrs**

Textbooks:

1. Computer Networks - A Top-down Approach, Andrew S. Tanenbaum, Fifth Edition, Pearson Education
2.TCP/IP Protocol Suite by B. A. Forouzan, TMGH Publication
3. Unix Network Programming – W. Rhichard Stevens Second Edition (PHI)
4. Linux User guide available on Internet (freeware).

Title of the Course: Mini Project Course Code: UCSE0451	L	T	P	Credit
			2	1

Course Pre-Requisite:
UBSH0208- Computer Programming
UBSH0238- Computer Programming Lab
UCSE0303-Data Structures
UCSE0333- Data Structures Lab

Course Description: Implementation of Mini Project using Programming Concepts.

Course Objectives: To expose students to:-

1. To identify the problem definition
2. To follow the methods and tasks of Software engineering
3. To utilize the techniques, skills and modern engineering tools necessary for building the project
4. To effectively demonstrate and present the ideas , methodology and technology used for the project

Course Learning Outcomes:

CO	After the completion of the course the student should be able to
CO1	Define the problem statement of the software project.
CO2	Organize an effective project plan with clear and finite objectives and document the synopsis and project reports.
CO3	Model the various modules of project with the help of DFDs , Flowcharts etc.
CO4	Develop the modules of proposed system.
CO5	Demonstrate the test cases for validation of proposed system.

CO-PO Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1		3											-	2
CO2		2								2	2		2	2
CO3			3		2								3	2
CO4			3						2			2	2	2
CO5				2									2	-

Assessments :

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 25 and 50 Marks respectively.

Marks	Assessment
25	ISE
50	ESE(POE)

Course Contents:

The mini project should be undertaken preferably by a group of 3-4 students who will jointly work and implement the project. The group will select a project with the approval of the guide and submit the name of the project with a synopsis, of the proposed work, of not more than 02 to 03 pages. The mini project should consist of defining the problem, analyzing, designing the solution and implementing it using a suitable programming language or tool. A presentation and demonstration based on the above work is to be given by the group. The work will be jointly assessed by a panel of teachers of the department. A hard copy of project report of the work done is to be submitted along with the softcopy of the project during ESE.

Rubrics for Evaluation

Sr. No.	Parameter	Unacceptable (E)	Marginal (D)	Adequate-Good (B+C)	Excellent (A)
1	Requirement Analysis	Irrelevant	Partially	Properly with few points left	Requirement analysis with all possible strategies defined
2	Design	No meaningful Design	Incomplete System Design	Presence of system design but no Proper Detailed Design	Presence of Correct system Design and Detailed Design
3	Coding & Testing	Code will not run	Code Runs Partially	Code runs with few errors or warnings	Code runs without errors for defined test cases
4	Report(Content)	Not proper	Relevant but no references and details	Content with relevant data and few spelling errors	Good Content with no spelling errors

Title of the Course: Soft Skills	L	T	P	Credit
Course Code:UCSE0461	-	-	2	-

Course Pre-Requisite:

Course Description: Soft skills are a combination of people skills, social skills, communication skills, character traits, attitudes, career attributes, social intelligence and emotional intelligence quotients among others that enable people to navigate their environment, work well with others, perform well, and achieve their goals with complementing hard skills.

Course Objectives:

12. Explain the importance of soft skills in corporate life.
13. Develop written skills of students to write corporate letters/emails.
14. Develop communication skills required for corporate etiquettes and ethics. 15. Develop presentation skills required for professional life.
16. Develop the ability to work in team.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloo 's level	's Cognitive Descriptor
CO1	Make use of effective communication skills in the corporate world.	3	Applying
CO2	Construct effective business letters/emails.	6	Creating
CO3	Demonstrate the corporate etiquettes and ethics	2	Understanding
CO4	Construct effective business presentations.	6	Creating
CO5	Work in team and show leadership skills.	2	Understanding

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO 1												1	1	
CO 2												1	1	
CO 3								2					2	
CO 4												1	1	
CO 5									2				1	

Assessments : Audit Course

Course Contents:

Unit 1: Art of communication Introduction to Soft Skills, Communication Theory, Effective Communication Skills, Barriers and Filters, Active Listening, Non Verbal Communication, Body Language.	**02 Hrs.**
Unit 2: Business Writing Skills	**03 Hrs.**

Business Letters/Emails - Format and Style, Types of Business Letter/Email – sales, order, complaint, adjustment, inquiry, follow-up, letter of recommendation, acknowledgement and resignation.	
Unit 3: World of teams Team concept, Elements of team work, Building an effective team, Role of Team Leader, Team based activities.	**02 Hrs.**
Unit 4: Adapting to corporate life Corporate Grooming and dressing Business Etiquette Business Ethics Dinning Etiquette Ethics policy.	**02 Hrs.**
Unit 5: Discussions, decisions and presentations What are group discussions, Types of Group Discussions, Corporate Presentations, Decision making, Resume Writing.	**03 Hrs.**
Unit 6: Job Interview: Types of Interviews -Telephonic, face to face, video, structured, unstructured, behavioral, problem solving, panel, Importance of body language.	**02 Hrs.**

Textbooks:

6. Personality Development and Soft- Skills , Barun K. Mitra ,Oxford University Press.
7. Business Communication : Making Connections in a Digital World 11th Edition (English, Paperback, Marie E. Flatley, Neerja Pande, Raymond V. Lesikar, Kathryn Rentz)

Unit wise Measurable students Learning Outcomes: Unit 1: Art of communication

UO1.1) To demonstrate the effective communication skills. UO1.2) To make use of appropriate body language.

Unit 2: Business Writing Skills

UO2.1) To interpret the importance of business writing skills.
UO2.2) To apply the appropriate business etiquettes in business letter/email.

Unit 3: World of teams

UO3.1) To explain the importance of team in corporate world. UO3.2) To demonstrate various team activities.

Unit 4: Adapting to corporate life

UO4.1) To demonstrate business etiquettes and ethics. UO4.2) To demonstrate corporate dressing.

Unit 5: Discussions, decisions and presentations UO5.1) To demonstrate group discussion activity. UO5.2) To apply presentation skills in a presentation.