Title of the Course: Discrete Mathematical Structures	**L**	**T**	**P**	**Credit**
Course Code: UITE0301	**3**	**1**	**-**	**4**

Course Pre-Requisite:
Basic knowledge of mathematics, set theory

Course Description:
To impart the necessary fundamental principles that is essential to study courses in computer science and related fields. To develop logical thinking and prerequisite knowledge necessary for skilled software engineer.

Course Objectives: To provide knowledge on
1.Mathematical logic
2.Set theory and operations on Set
3. Functions and relations
4.Basic terminologies of graph theory
5.Algebriac system and application

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	Construct and minimize different Mathematical Logical equation.	II	Understanding
CO2	Construct various problem based on Sets, relations and functions.	VI	Creating
CO3	Summarize the Graph theory and its applications	II	Understanding
CO4	List different Algebraic systems and its applications.	I	Remembering

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**	**PSO1**	**PSO 2**
CO1	3	3											1	
CO2	3	3	2											2
CO3	3		2											
CO4	3		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 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:Mathematical Logic Statements and notation, Connectives, Conditionals and Bi conditionals, well formed formulas, tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal forms, Theory of Inference for statement calculus – validity using truth table, rules of inference, consistency of Premises and indirect method of proof.	**8Hrs.**
Unit 2:Set Theory Basic concepts of set theory-Notation, Inclusion and Equality of Sets, The power set, Some Operations on Sets, Venn Diagram, Ordered Pairs and n-tuples, Cartesian Products, Permutations, combinations, Discrete Probability	**8 Hrs.**
Unit 3:Relations and Functions **Relations and Ordering**:-Relations, Properties of Binary Relations in a Set, Relation Matrix and Graph of Relation, Equivalence relations, Partial Ordering and Partial Ordering Set **Functions:-** Definition and Introduction, , Composition of functions, Inverse Functions, recursive functions	**8 Hrs.**
Unit 4: Algebraic systems Algebraic Systems, Semi Groups, Groups, Monoid, Abelian Groups, subgroups, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups	**6 Hrs.**
Unit 5: Lattices and Boolean algebra Lattice as POSETs , definition , examples and properties, Lattice as algebraic systems, Special lattices, Boolean algebra definition and examples, Boolean functions, representation and minimization of Boolean functions.	**6 Hrs.**
Unit 6: Graph theory Basic concepts of graph theory, Storage representation and manipulation of Graphs, PERT and related techniques.	**6 Hrs.**
Textbooks: 1. Discrete Mathematical Structures with Application to Computer Science - J. P. Tremblay & R. Manohar (MGH International) 2. Elements of Discrete Mathematics- C. L. Liu and D. P. Mohapatra,, 4Edition McGraw-Hill(unit 4)	

References:

1] Discrete Mathematics - Semyour Lipschutz, MarcLipson (MGH), Schaum's outlines
2] Discrete Mathematics and its Applications - Kenneth H. Rosen (AT&T Bell Labs) (mhhe.com/rosen)

.

Unit wise Measurable students Learning Outcomes:

UO1.1 Construct and minimize different Mathematical Logical equation
UO2.1 Construct problem based on Sets
UO 3.1 Solve various problem based on relations and functions
UO 4.1 Analyze various algebraic system and its applications
UO 5.1 Analyze lattices and Boolean algebra system and its applications
UO 6.1Summarize the basic concepts of graph theory

Title of the Course: Applied Mathematics	L	T	P	Credits
Course Code: UITE0302	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 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 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	Bloom's Cognitive level	Descriptor
CO1	**Explain** fuzzy numbers in dealing with fuzzy equations.	II	Understanding
CO2	**Explain** the fuzzy sets and fuzzy logic in dealing with real problems.	II	Understanding
CO3	**Solve** algebraic and transcendental equations by using numerical methods**.**	III	Applying
CO4	**Make use of** method of least squares to fit the curves for given bivariate data.	III	Applying
CO5	**Apply** the knowledge of the probability distributions to **solve** problem arising in engineering.	III	Applying
CO6	**Simplify** the systems of simultaneous linear equations and **solve** them by using Gauss-Jordan method and LU decomposition method.	IV	Analyzing

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2										1
CO2	3	2										1
CO3	3	2										1
CO4	3	2										1
CO4	3	2										1
CO5	3	2										1
CO6	3	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:--- Linear algebra 1. Solutions of simultaneous linear equations using Gauss-Jordan method. 2. Solutions of simultaneous linear equations using LU decomposition method. 3. Determination of Eigen Value by Iteration method. 4. Solution of non-linear simultaneous equations.	5 **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.	7 **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 of fuzzy set. 2. Basic concepts of fuzzy sets.	7 **Hrs.**

3. Basic operations on fuzzy sets. 4. Properties of fuzzy sets. 5. Multivalued Logics. 6. Inference from conditional fuzzy propositions.	
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.	7 **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:--- Linear algebra** UO 1.1 Evaluate solutions of simultaneous linear and non linear equations. UO 1.2 Determine of Eigen Value by Iteration method. **Unit 2:--- Numerical methods.** UO 2.1 Solve numerical integration problems. UO 2.2 Understand special numerical integral formulae. UO 2.3 Solve numerically transcendental and algebraic equations. **Unit 3:--- Probability and Distributions.** UO 3.1 Define the concept of random variable. UO 3.2 Solve the function as probability density function. UO 3.3 Solve Binomial, Poisson and Normal distribution problems. **Unit 4:--- Statistical Techniques** UO 4.1 Understand correlation Coefficient. UO 4.2 Apply fitting of curves for bivariate data. UO 4.3 Make use of Testing of Hypothesis. **Unit 5:--- Introduction to Fuzzy sets and Fuzzy Logic** UO 5.1 Understand Basic concept of Fuzzy set theory. UO 5.2 Define membership functions. UO 5.3 Apply Basic operations on Fuzzy set. UO 5.4 Apply of Fuzzy set properties to interpret. **Unit 6:--- Fuzzy Arithmetic** UO 6.1 Apply Fuzzy numbers and Fuzzy cardinality. UO 6.2 Apply operate arithmetic operations on fuzzy numbers. UO 6.3 Solve fuzzy equations.	

Title of the Course: Data Communication and Networks	L	T	P	Credit
Course Code: UITE0303	4	--	--	4

Course Pre-Requisite: Fundamentals of Computers

Course Description:
The course gives you fundamental knowledge of data communication networks including its components. It also covers physical layer, data link layer and network layer in depth knowledge. Real world framing, error correction and detection, flow control techniques are also covered with numerical. Importantly socket programming basics is included to implement some above mentioned techniques.

Course Objectives:

1. Explain types of networks, topologies and networks models
2. Explain different types of data encoding techniques
3. Understand concept of multiplexing and switching
4. Understand functions of data link layer
5. Understand functions of network layer
6. Explain fundamentals of socket interfaces for client-server communication

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	Explain different network topologies, multiplexing ,switching data encoding techniques	II	Understanding
CO2	Demonstrate working of different interconnecting devices using simulation tools	II	Understanding
CO3	Solve numerical based upon signal transmission impairment	III	Applying
CO4	Compare different congestion control and routing algorithms	IV	Analyzing
CO5	Design program for framing, flow control and error correction and detection techniques using socket programming	VI	Creating

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2													
CO2			2											
CO3			1											
CO4						2								
CO5			3										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 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: Data Communication Fundamentals Data Communication – Definition, Components, Data representation, Data Flow Networks – Definition, Uses, Topologies, Categories, Internet – History, ISP hierarchy, Protocols & Standards – Protocols, Standards, Standards Organizations, Network Models.	**7 Hrs.**
Unit 2 : Data Signals and Data Encoding Analog & Digital data, Analog & Digital signals, Transmission Impairments, Data Rate Limits and Performance. Analog-to-Digital conversion –PCM, DM Digital-to-Analog conversion – ASK, FSK, PSK .Analog-to-Analog conversion – AM, FM, PM	**12 Hrs.**
Unit 3 : Multiplexing & Switching Parallel and serial transmission, Asynchronous and Synchronous transmission, Multiplexing–FDM, WDM, TDM. Switching –Circuit switched, Packet switched, Message switched, Structure of switches	**07Hrs.**
Unit 4 : Data Link Layer Framing, Error Control, Flow Control, Error detection & correction codes Elementary data link protocols- Simplex, Stop & Wait, Simplex for noisy channel. Sliding window protocols – 1-bit, go back n, selective repeat Channel allocation- static, dynamic Multiple access protocols: Aloha, CSMA, Collision Free Protocols IEEE 802 Standards for LAN and MAN – 802.3, 802.4, 802.5	**11 Hrs.**
Unit 5 : Network Layer IPv4 Addresses: Introduction, Classful and Classless Addressing, Special Addresses, Network Layer Design Issues Routing Algorithms **:** Shortest Path, Flooding, Distance Vector, Link State, Broadcast Congestion control algorithms: Principles, Congestion prevention policies, Traffic Shaping, congestion control in datagram subnet, Choke Packet, Load Shedding, Jitter Control	11 **Hrs.**
Unit 6 : Berkeley Sockets Socket Addresses, Elementary Socket system calls byte ordering and address conversion routines, connectionless iterative server, Connection Oriented concurrent server, TCP and UDP Client server Programs	**08 Hrs.**

Textbooks:

5. Data Communications and Networking –Forouzon ,5thEdition ,TMGH**.(1,2,3)**
6. Computer Networks – A. S. Tannenbaum.,3rd Edition,PHI**.(4,5)**
7. Unix Network Programming , W Richard Stevens, PHI**.(6)**

References:

1) Data Communication & Networks: An Engineering Approach by Irvine, Wiley India Ltd.
2) TCP/IP protocol suite, B A Forouzan, TMGH.
3) Computer Networks: Principles ,Technologies and Protocols for Network Design, Wiley

Unit wise Measurable students Learning Outcomes:

Unit 1: Data Communication Fundamentals

UO 1.1 State and define basic concepts of data communication

UO 1.2 Identify different types of networks and topologies

UO 1.3 Explain various concepts about internet

UO 1.4 Enlist different protocols and standards

Unit 2 : Data Signals and Data Encoding

UO 2.1 Define and distinguish analog and digital signals

UO 2.2 Explain different transmission impairments and data encoding

UO 2.3 compare different data encoding techniques

UO 2.4 State and describe various applications of data encoding

Unit 3: Multiplexing & Switching

UO 3.1 List and explain different data transmission modes

UO 3.2 Describe different multiplexing techniques

UO 3.3 Explain the structures of various switching devices

UO 3.4 Explain various types of switched networks

Unit 4 : Data Link Layer

UO 4.1 Implement various DLL functions

UO 4.2 Compare different data link protocols

UO 4.3 Differentiate between channel allocation schemes

Unit 5: Network Layer

UO 5.1 Identify different types of IP addresses

UO 5.2 Compare different routing algorithm

UO 5.3 Differentiate congestion control algorithm

UO 5.4 Explain network layer design issues

Unit 6: Berkeley Sockets

UO 6.1 List different socket system calls UO 6.2 Explain working of client-server using socket system call UO 6.3 Design client-server program using different socket calls

Title of the Course: Digital Systems and Microprocessors	**L**	**T**	**P**	**Credit**
Course Code:UITE0304	**3**	**-**	**-**	**3**

Course Pre-Requisite: Fundamentals of Electronics and Computers, basic number system

Course Description: This course is aim to Learn various digital system concepts and microprocessors

Course Objectives:

CLO-1: To learn basic digital design techniques.
CLO-2: To design and construction of combinational and sequential circuits.
CLO-3: To study the architecture & working of 8086 microprocessor and peripheral.
CLO-4: To learn the assembly language programming 0f 8086 Microprocessor.
CLO-5: To Study NDP (8087 microprocessor).

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		level	**Descriptor**
CO1	Solve different examples of arithmetic and logical operations on various number systems.	3	Applying
CO2	Design and demonstrate different sequential and combinational-logic design.	3	Applying
CO3	Summarize the working of 8086 microprocessor and peripheral.	2	Understanding
CO4	Design and execute assembly language programs using 8086 instruction set.	3	Applying
CO5	Distinguish different instructions using timing diagrams.	4	Analyzing

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3												1	
CO2	2	2	1										2	
CO3	2												2	
CO4			2		2	1			1					3
CO5		2												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 with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Logic Design Minimization Techniques Introduction to number system and logic gates. **Boolean algebra:** Truth tables and Boolean algebra. Idealized logic gates and symbols. DeMorgan's rules Axiomatic definition of Boolean algebra, Basic theorems and properties of Boolean algebra. **Logic minimization:** Representation of truth-table, SOP form, POS form, Simplification of logical functions, Minimization of SOP and POS forms, expansion of Boolean expression (standard SOP & POS). **Reduction techniques:** K-Maps up to 4 variables, prime Implicant, Don't care Conditions.	**6 Hrs.**
Unit 2:Combinational Logic **Codes:** BCD, Excess-3, Gray code , Binary Code , BCD addition & subtraction. **Circuits:** Half- Adder, Full Adder, Half Subtract or, Full Sub tractor, BCD adder using and subtract using 7483 **Multiplexers (MUX):** Working of MUX, Implementation of expression using MUX (IC 74153, 74151). **De-multiplexers (DEMUX):** Working of DEMUX, Implementation of expression using DEMUX, Decoder. (IC 74138). BCD to 7 segment decoder	**5 Hrs.**
Unit 3: Sequential Logic **Introduction:** Sequential Circuits. Difference between combinational circuits and sequential circuits **Flip-flop:** SR, JK, D, T; Preset & Clear, Master and Slave Flip Flops their truth Tables and excitation tables, Conversion from one type to another type of Flip Flop. Application of Flip-ops: Bounce Elimination Switch, registers, counters. Registers: Buffer register; shift register. **Counters:** Asynchronous counter. Synchronous counter, ring counters, BCD Counter, Johnson Counter, Modulus of the counter (IC 7490)	**7 Hrs.**
Unit 4: Introduction to 8086 Evolution of microprocessors, Introduction to 16 bit microprocessor, Architecture and Pin diagram of 8086, Programmers model of 8086 (Registers), Segmentation, logical to physical address translation, Read write cycle timing diagrams, Address mapping and decoding, I/O: memory mapped I/O & I/O Mapped I/O.	**6 Hrs.**
Unit 5: 8086 Instruction set and interrupts Structure Addressing modes, Instruction set of 8086 in detail, Instruction Formats, Stacks, Assembly Language Programming, Assembler, Linker, Debugger (Turbo debugger), Directives, Procedures (Near & Far), Macros, Loop constructs, 8086 Programming examples. 8086 Interrupt Structure, Interrupt Vector Table (IVT), ISR, Hardware and software Interrupts	**10 Hrs.**
Unit 6:Minimum & Maximum mode of 8086 and Introduction to 8087	**6 Hrs.**

Minimum & Maximum mode of 8086: Multifunction pins of 8086, 8088 bus controller, IOB mode of 8288, Minimum & Maximum mode con_guration diagram 8087(NDP) - Features, Block Diagram, Data Types, Control & status registers, typical Instruction Set & Programming	

Textbooks:

1. Fundamental of Digital Circuits {A. Anand Kumar, 2nd Edition, PHI Private Limited.
2. Douglas Hall, \Microprocessors & Interfacing", McGraw Hill, Revised 2nd Edition, 2006 ISBN 0-07-100462-9
3. John Uffenbeck," The 8086/88 Family: Design, Programming & Interfacing", PHI,

References:

1. R. Jain, \Modern Digital Electronics", eighth edition, Tata McGraw-Hill, 2003, ISBN 0 { 07 { 049492 { 4
2. A. Ray, K. Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing", Tata McGraw Hill,2004 ISBN 0-07-463841-6
3. Liu, Gibson, \Microcomputer Systems: The 8086/88 Family", 2nd Edition, PHI,2005

Unit wise Measurable students Learning Outcomes:

Unit 1:Logic Design Minimization Techniques

UO1.1: Acquire basic knowledge of logic gates which are used to design the circuit
UO1.2: Solve the Boolean expression to minimize the logic gate in designing the circuit

Unit 2: Combinational Logic

UO2.1: Describe various combinational circuits and their use
UO2.2: Design any combinational circuit for given equation and truth table

Unit 3: Sequential Logic

UO3.1: differentiate combinational and sequential circuits
UO3.2: List and explain working of various flip-flop
UO3.3: Construct different sequential circuits

Unit 4: Introduction to 8086

UO4.1: Describe the components of 8086 microprocessor
UO4.2: Explain 8086 microprocessor architecture

Unit 5: 8086 Instruction set and interrupt Structure

UO5.1: Apply different instructions for assembly language programming
UO5.2: Execution of various assembly language programs

Unit 6:Minimum & Maximum mode of 8086 and Introduction to 8087

UO6.1: Explain Minimum & Maximum mode of 8086
UO6.2: Describe the components of 8087 microprocessor

Title of the Course: Data Structure	**L**	**T**	**P**	**Credit**
Course Code:UITE0305	**3**	**-**	**-**	**3**

Course Pre-Requisite: Fundamentals of Programming Language C

Course Description: This course is aim to Learn various data structures that are used to store and organize data.

Course Objectives:

1. To understand the different ways of data representation.
2. To define high level of abstraction needed for data structure and algorithm.
3. To study the representation, implementation and applications of linear and non linear data structures.
4. Compute the complexity of various algorithms.
5. To develop application using data structure algorithms.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		level	**Descriptor**
CO1	Explain the basic concepts of Data Structures.	2	Understanding
CO2	Apply various data structures to solve different computing problems.	3	Applying
CO3	Choose appropriate data Structure that efficiently model the information in a problem.	5	Evaluating
CO4	Analyze searching and sorting algorithms to find their complexity.	4	Analyzing

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3				1								1	
CO2		2	2	2		2							1	3
CO3		2		2		2							1	2
CO4		2	2	2		1							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 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 to Data structures Concept of data, Data object, Data structure, Abstract Data Types (ADT), Concept of Primitive and non primitive, linear and Non-linear, pseudo code, algorithm efficiency, File structure	**04 Hrs.**
Unit 2:Linear Data Structures Linked Organization Limitations of static memory allocation. Dynamic memory allocation in C. Concept of linked organization, Singly linked list, Doubly linked list, Circular linked list. Operations like insertion, deletion, traversal & other operations on these data structures,Garbage collection and compaction **Applications:** Josephus Problem**,** Representation and manipulation (addition) of polynomial (implementation not expected)	**08 Hrs.**
Unit 3: Linear Data Structure : Stacks and Queues **Stack:** Concept of stack as ADT, Representation and implementation of stack using sequential & linked organization. **Applications:** Simulating recursion using explicit stack, Arithmetic expression conversion: infix to postfix ,Evaluation of postfix expression, reversing a string, parsing : well- formed parenthesis checking **Queue:** Concept of queue as ADT, Representation and implementation of linear queue & circular queue using sequential & linked organization, Double ended queue, Multi-queue and Priority queue. **Applications**: Job scheduling, Queue simulation, Categorizing data (Implementation not expected)	**06 Hrs.**
Unit 4: Non Linear Data Structure : Trees Basic Concept, Terminology and user representation of Trees, Binary Trees, Algorithms for Binary Tree Traversal, Binary search trees (BST), algorithms on BST, AVL Trees, Heap Tree, B Tree, B+ Trees **Applications:** Evaluating the expression, infix to prefix and Infix to postfix conversion using tree	**08 Hrs.**
Unit 5:Non Linear Data Structure : Graph Concepts and terminology of graph, Representation of graph using adjacency matrix and adjacency list, Elementary Graph Operations: Depth first Search, Breath first search, Spanning Trees	**07 Hrs.**

Applications: Minimum spanning Tree: Prim's and Kruskal's Algorithm, Shortest Path: Single- Source (Dijkstra's Algorithm), All-Pairs Shortest Path (Floyd's Warshall Algorithm)	
Unit 6:Searching & Sorting Techniques and Hashing Need of sorting and searching, sorting order & stability in sorting. **Sorting Techniques:** Concept of Internal & External sorting, Algorithms for Bubble sort, Selection sort, Insertion sort, Radix sort, Heap sort, Quick sort and Merge sort. Analysis of each sorting technique for best, worst and average case, **Searching Techniques:** Algorithms for Sequential search, Binary search, analysis of each searching technique for best, worst and average case. Hashing Techniques, Types of Hash Functions, Collision resolution techniques, open and closed hashing	**09 Hrs.**
Textbooks: 1. R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning, ISBN 9788131503140(Refer 1,3,4,5). 2.Seymour Lipschutz,G.A.V.Pai ,"Data Structure",Tata McGraw Hill,ISBN-13:978-0-07-060168-0.(Refer Chapter:6) 3.Rohit Khurana,"Data Structures Using C",Vikas publishing House Pvt.ltd,ISBN:978-93259-7565-1.(Refer Chapter:2 and application of DS)	
References: 1.E.Horowitz,S.Sahani,S.Anderson-Freed "Fundamentals of Data Structures in C", Universities Press ,2008 ,ISBN 10:8173716056.	
Unit wise Measurable students Learning Outcomes: **Unit 1: Introduction to Data structures** UO1.1: Define the importance of structure and abstract data type, and their basic usability in different applications through different programming languages **Unit 2: Linear Data Structures Linked Organization** UO2.1: Explain the linked implementation, and its uses both in linear and non-linear data structure. UO2.2: Implement linked list data structure to solve various problems. **Unit 3: Linear Data Structure : Stacks and Queues** UO3.1: Define various data structure such as stacks, queues. UO3.2: Solve problems using data structures such as stacks, queues. **Unit 4: Non Linear Data Structure : Trees** UO4.1: define different tree traversal techniques. UO4.2: Apply tree data structure to solve various problems. **Unit 5: Non Linear Data Structure : Graph** UO5.1: Implement different graph traversal techniques. UO5.2: Solve problems using graphs such as minimum spanning trees, shortest path algorithm. **Unit 6:Searching and Sorting Techniques** UO6.1: Implement various kinds of searching and sorting techniques UO6.2: Assess how the choice of data structures and algorithm design methods impacts the performance of programs. UO6.3: Solves problems using hashing techniques	

Course Name: **Data Structures**

Course Code: **UITE0305**

Problem Statements:

DSPBLPB01*:* Problem Statement

There is a car agency that contains different kinds of cars. They may have more than one car from each kind. They want an application to maintain the record of customer and sold car. Each customer gives an order to buy a car. Car agency search for availability of car and sold the car on first come first serve basis. Also, car agency gives the car on rent. Rent of car will be calculated on distance covered between source and destination. Car agency want to calculate minimum rent and give the suggestion of route. How you are going to develop an application to solve all the requirement of car agency?

2. Activities with timeline:

Sr. No.	Activity	Timeline
1	PBL awareness in class	1st week
2	Announcement of problem/s for PBL	2nd week
3	Team formation	3rd week
4	Project ISE I:Synopsis presentation	5th week
5	Completion of corrections/improvements in synopsis	6th week
6	Project ISE II: Project Progress Presentation with Model/case study	10th week
7	Completion of correction/improvements in Evaluation II	11th week
8	End Semester Evaluation of Project	13th week
9	Determining future scope for improvement	14th week

3. Assessment Scheme:

- ISE-I
- ISE-II
- Project ESE

4. Evaluation Scheme:

- **Project ISE I :** Synopsis presentation for 5 marks (evaluation with rubrics)
- **Project ISE II:** Project Progress Presentation with Model/case study for 5 marks (evaluation with rubric)
- **End Semester Evaluation of Project:** Multimedia presentation and demonstration of working models for 15 marks out of 25 of course lab ISE (evaluation with rubrics)

Title of the Course: Data Communication and Networks Lab	L	T	P	Credit
Course Code: UITE0331	--	--	2	1

Course Pre-Requisite: Data Communication and Networks Concepts , Computer Programming

Course Description:
The course gives you fundamental knowledge of data communication networks including its components. It also covers physical layer, data link layer and network layer in depth knowledge. Real world framing, error correction and detection, flow control techniques are also covered with numerical. Importantly socket programming basics is included to implement some above mentioned techniques.

Course Objectives:

1. Explain types of networks, topologies and networks models
2. Explain different types of data encoding techniques
3. Understand concept of multiplexing and switching
4. Understand functions of data link layer
5. Understand functions of network layer
6. Explain fundamentals of socket interfaces for client-server communication

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	Explain different network topologies, multiplexing ,switching data encoding techniques	II	Understanding
CO2	Demonstrate working of different interconnecting devices using simulation tools	II	Understanding
CO3	Solve numerical based upon signal transmission impairment	III	Applying
CO4	Compare different congestion control and routing algorithms	IV	Analyzing
CO5	Design program for framing, flow control and error correction and detection techniques using socket programming	VI	Creating

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	PSO 2
CO1	2													
CO2			2											
CO3			1											
CO4						2								
CO5			3										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	50
ESE	50

ISE are based on assignment/declared test/quiz/seminar/Group Discussions etc.
ESE: Assessment is based on practical oral examination.

Course Contents:

Experiment No. 1 : Demonstration of Networks and Components **Aim :** Demonstration of Networks and Components **Objective :** To study and demonstrate the different interconnecting devices of LAN and WAN **Steps to Perform Experiment :** 1. Refer DC Tutorial No. 1 : KIT Campus Network Visit and list out different interconnecting devices and components of Campus Network 2. Design the Campus network using Cisco Packet Tracer 3. Simulate the working of every interconnecting devices in same network using Cisco Packet Tracer List of interconnecting devices and components but not limited to : Hub,Switch,Bridge,Router,Firewall,Server **Outcome :** After this experiment student will able to explain able to describe the working and use of each interconnecting devices and components.	**4 Hrs**
Experiment No. 2 : Connectivity Tools **Aim:** Study of Connectivity Tools **Objective :** Demonstrate the use of different connectivity tools **Tools:** **a) ifconfig** Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface. ifconfig eth0- display the current status of interface mentioned in command ifconfig -a - display status of all the available interfaces on the computer ifconfig eth0 down- shutdown the interface mentioned in command (super user privileges are required) ifconfig eth0 up- up the interface mentioned in command (super user privileges are required) ifconfig 172.25.3.5 - set the given ip address to the interface **b) arp** Arp manipulates the kernel's ARP cache in various ways. The primary options are clearing an address mapping entry and manually setting up one. For debugging purposes,	**2 Hrs**

the arp program also allows a complete dump of the ARP cache. arp -a -show the entries of specified host arp -s ip addr mac address- used to add corresponding entry in cache arp -d ip addr - used to delete specific entry from the cache **c) route** route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig(8) program. When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables. route - display kernel's IP routing tables **d) traceroute** traceroute tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's time to live (TTL) field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to the host. traceroute 210.212.172.190 - displays the response from each gateway. **e) nmap** Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. nmap 172.25.3.100 - scanning the given system nmap 172.25.3.100 172.27.100.2 - scanning two systems **f) netstat** Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. Netstat prints information about the Linux networking subsystem. netstat- display network subsystem information **g) finger** The finger displays information about the system users finger -s - Finger displays the user's login name, real name, terminal name and write status idle time, login time, office location and office phone number. **Outcome:** Student will be able to execute command and show the outputs according to different options	
Experiment No. 3 : Socket Interfaces for Client Server Models **Aim:** Study of Socket Interfaces for Client Server Models **Objective:** Demonstrate use of different socket interfaces **Socket System Calls:** **What is socket programming?** Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket (node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server. **Stages for server**	**4 Hrs**

▪ **Socket creation:**
int sockfd = socket(domain, type, protocol)
sockfd: socket descriptor, an integer (like a file-handle)
domain: integer, communication domain e.g., AF_INET (IPv4 protocol) , AF_INET6 (IPv6 protocol)
type: communication type
SOCK_STREAM: TCP(reliable, connection oriented)
SOCK_DGRAM: UDP(unreliable, connectionless)
protocol: Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.(man protocols for more details)
▪ **Setsockopt:**
▪ int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);
This helps in manipulating options for the socket referred by the file descriptor sockfd. This is completely optional, but it helps in reuse of address and port. Prevents error such as: "address already in use".
▪ **Bind:**
▪ int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure). In the example code, we bind the server to the localhost, hence we use INADDR_ANY to specify the IP address.
▪ **Listen:**
int listen(int sockfd, int backlog);
It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The backlog, defines the maximum length to which the queue of pending connections for sockfd may grow. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.
▪ **Accept:**
int new_socket= accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer data.
Stages for Client
▪ **Socket connection:** Exactly same as that of server's socket creation
▪ **Connect:**
▪ int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr. Server's address and port is specified in addr.
 	
Outcome: Student will be able to use socket system call for implementing client-server model	

Experiment No. 4 : Network Protocol Analyzer **Aim :** To study Network Protocol Analyzer **Objective:** Demonstrate use of NPA Wire Shark for analysis of network traffic **Network Protocol Analyzer i.e. Wire shark** https://www.lifewire.com/wireshark-tutorial-4143298 **Outcome:** Student will be able to use network protocol analyzer for monitoring network traffic	**2Hrs**

Experiment No. 5 : Framing Methods **Aim:** Study of Framing Methods **Objective:** Design and implement character count and bit stuffing technique **Character Count:** Student is required to write a sender program where sender will take actual data to be transmitted as input from user and store it in buffer. - Now character count is added at the beginning of every frame. (You can also display the codeword) - Write a receiver program where it accepts the codeword sent from sender end. (You can also display the accepted codeword from sender) - Now check count and bits from the codeword and retrieve the actual data. Now display the actual data **Bit Stuffing :** Student is required to write a sender program where sender will take actual data to be transmitted as input from user and store it in buffer. - Now redundant 0 bit has to be stuffed after every 5 consecutive 1's in the actual data and a codeword has to be created which has to be sent to receiver. (You can also display the codeword) - Write a receiver program where it accepts the codeword sent from sender end. (You can also display the accepted codeword from sender) - Now destuff (remove) the redundant 0 bits from the codeword and retrive the actual data. Now display the actual data. **Outcome:** Student will be able to implement and demonstrate working of the same.	**4 Hrs**
Experiment No. 6 : Error Correction and Detection **Aim:** To study Error Correction and Detection techniques **Objective:** Design and implement Hamming Code technique **Outcome:** Student will be able to implement and demonstrate working of the Hamming code	**2 Hrs**
Experiment No. 7 : Sliding Window **Aim:** To study Sliding Window Technique **Objective:** Design and implement Stop and Wait Protocol Design and implement GO BACK N Protocol Design and implement Selective Repeat Protocol **Outcome:** Student will be able to implement and demonstrate working of the Sliding Window techniques	**4 Hrs**
Experiment No. 8 : Routing Algorithm **Aim:** To study Routing Algorithm **Objective:** Design and implement STP routing protocol **Outcome:** Student will be able to implement and demonstrate working of the Sliding Window techniques	**2 Hrs**
Textbooks: **1.** Data and Computer Communications –Williams Stallings ,5thEdition ,PHI**.(1,2,3)** **2.** Computer Networks – A. S. Tenebaum.,3rd Edition,PHI**.(4,5)** **3.** Unix Network Programming , W Richard Stevens, PHI**.(6)**	
References: 1) Data Communication & Networks: An Engineering Approach by Irvine, Wiley India Ltd. 2) TCP/IP protocol suite, B A Forouzan, TMGH.	

3) Computer Networks: Principles ,Technologies and Protocols for Network Design, Wiley

Unit wise Measurable students Learning Outcomes:

Unit 1: Data Communication Fundamentals

UO 1.1 State and define basic concepts of data communication

UO 1.2 Identify different types of networks and topologies

UO 1.3 Explain various concepts about internet

UO 1.4 Enlist different protocols and standards

Unit 2 : Data Signals and Data Encoding

UO 2.1 Define and distinguish analog and digital signals

UO 2.2 Explain different transmission impairments and data encoding

UO 2.3 compare different data encoding techniques

UO 2.4 State and describe various applications of data encoding

Unit 3: Multiplexing & Switching

UO 3.1 List and explain different data transmission modes

UO 3.2 Describe different multiplexing techniques

UO 3.3 Explain the structures of various switching devices

UO 3.4 Explain various types of switched networks

Unit 4 : Data Link Layer

UO 4.1 Implement various DLL functions

UO 4.2 Compare different data link protocols

UO 4.3 Differentiate between channel allocation schemes

Unit 5: Network Layer

UO 5.1 Identify different types of IP addresses

UO 5.2 Compare different routing algorithm

UO 5.3 Differentiate congestion control algorithm

UO 5.4 Explain network layer design issues

Unit 6: Berkeley Sockets

UO 6.1 List different socket system calls

UO 6.2 Explain working of client-server using socket system call

UO 6.3 Design client-server program using different socket calls

Title of the Course: Digital Systems and Microprocessors Lab	L	T	P	Credit
Course Code: UITE0332	-	-	2	1

Course Pre-Requisite: Fundamentals of Electronics and Computers, Instruction set of 8086

Course Description:

DSM Lab deals with study and design combinational and sequential circuits using logic gates, and writing programs using 8086 instructions

Course Objectives:

1. To learn basic digital design techniques.
2. To learn the assembly language programming 0f 8086 Microprocessor
3. To understand design and construction of combinational and sequential circuits.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Design any circuits using logic gates	2	Understanding
CO2	Design and apply combinational and sequential circuits	3	Applying
CO3	Write and execute assembly language programs using 8086 instructions	4	Creating

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	PSO1	PSO 2
CO 1	3		1										2	
CO 2			2											1
CO 3		2		1		2								3

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 oral examination

Course Contents:

Experiment No. 1: STUDY OF ALL LOGIC GATES **Aim and Objectives:** To familiarize basic logic gates and universal gate ICs, and verify its truth table **Outcomes:** Verification of truth table of all logic gates **Theoretical Background:** use of Digital Trainer Kit, Working of different logic gates and there IC diagrams **Experimentation:** 1. Place the ICs on trainer kit.	**2 Hrs.**

2. Wire the circuit diagram 3. Connect the inputs to the input switches provided in the trainer kit. 4. Connect the outputs to the terminals of output LEDs. 5. Connect VCC and GND to respective pins of trainer kit 6. Apply various combinations of inputs according to the truth table and observe condition of LEDs **Results and Discussions**: Familiarize the digital IC trainer kit & logic gate IC packages and verified the truth tables of logic gates **Conclusion:**	
Experiment No. 2: STUDY OF COMBINATIONAL CIRCUITS **Aim and Objectives:** To verify De-Morgan's theorem for two variables **Outcomes:** De-Morgan's theorem verified **Theoretical Background:** De-Morgan's theorem Statement, IC'c required. **Experimentation:** 1. Place the ICs on trainer kit. 2. Wire the designed circuit diagram 3. Connect the inputs to the input switches provided in the trainer kit. 4. Connect the outputs to the terminals of output LEDs. 5. Connect VCC and GND to respective pins of trainer kit 6. Apply various combinations of inputs according to the truth table and Observe the output. **Results and Discussions:** De-Morgan's theorems of Boolean algebra were verified. **Conclusion:**	**2 Hrs.**
Experiment No. 3: HALF ADDER AND HALF SUBTRACTOR **Aim and Objectives:** To design and set up half adder and half subtractor using a. EXOR gates and AND gates b.NAND gates **Outcomes:** Half adder and the half subtractor circuits are set up using logic gates and verified the result **Theoretical Background:** Circuit diagram and truth table of half adder and subtractor, IC's required. **Experimentation:** 1. Place the ICs on trainer kit. 2. Wire the circuit diagram 3. Connect the inputs to the input switches provided in the trainer kit. 4. Connect the outputs to the terminals of output LEDs. 5. Connect VCC and GND to respective pins of trainer kit 6. Apply various combinations of inputs according to the truth table and observe the outputs. **Results and Discussions:** Half adder and the half subtractor circuits are set up using logic gates and verified the result **Conclusion:**	**2 Hrs.**
Experiment No. 4: BCD to EXCESS -3 CODE CONVERTER **Aim and Objectives:** To design and set up the circuit of BCD to Excess-3 converter **Outcomes:** the circuit of BCD to Excess-3 converter has set up and verified the result	**2 Hrs.**

Theoretical Background:knowledge of BCD and Excess-3 code. **Experimentation:** 1. Place the ICs on trainer kit. 2. Wire the circuit diagram 3. Connect the inputs to the input switches provided in the trainer kit. 4. Connect the outputs to the terminals of output LEDs. 5. Connect VCC and GND to respective pins of trainer kit 6. Apply various combinations of inputs according to the truth table and observe the outputs. **Results and Discussions:** the circuit of BCD to Excess-3 converter has set up and verified the result **Conclusion:**	
Experiment No. 5: FLIP FLOPS USING GATES AND FAMILIARIZATION OF ICs **Aim and Objectives:** To Setup the following flip flops using gates and verify the truth table also familiarize the flip flop ICs 1.GatedRS flip flop 2.JK flip flop **Outcomes:** The flip flops, Gated RS flip flop , JK flip flop and were set up using NAND gates and verified **Theoretical Background:** Circuit diagram and truth table of RS flip flop, JK flip flop , IC's required. **Experimentation:** 1. Place the ICs on trainer kit. 2. Wire the circuit diagram 3. Connect the inputs to the input switches provided in the trainer kit. 4. Connect the outputs to the terminals of output LEDs. 5. Connect VCC and GND to respective pins of trainer kit 6. Apply various combinations of inputs according to the truth table and observe the outputs. **Results and Discussions:** The flip flops, Gated RS flip flop , JK flip flop and were set up using NAND gates and verified. ICs 7476 is familiarized **Conclusion:**	**2 Hrs.**
Experiment No. 6: ASYNCHRONOUS COUNTERS **Aim and Objectives:** To set up and study the working of asynchronous up counter and down counter. **Outcomes:** Asynchronous up counter and down counter is designed and truth table is verified. **Theoretical Background:** Circuit diagram and truth table of Asynchronous up and down counter, IC's Required. **Experimentation:** 1. Place the ICs on trainer kit. 2. Wire the circuit diagram 3. Connect the inputs to the input switches provided in the trainer kit. 4. Connect the outputs to the terminals of output LEDs. 5. Connect VCC and GND to respective pins of trainer kit 6. Apply various combinations of inputs according to the truth table and Observe the outputs. **Results and Discussions**: Asynchronous up counter and down counter is designed and truth table is verified. **Conclusion:**	**2 Hrs.**

Experiment No. 7: Assembly language Programs using 8086 instructions **Aim and Objectives:** To write an assembly language program for 1. Addition, Subtraction, multiplication of two 16/32 bit numbers. 2. to find the sum of no.s stored in array 3. display message on screen using macros 4. to find the factorial of no.s 5. reverse the word in string 6. display 4 digit BCD no. on 7-segment display using display inteface. 7. to sort the given numbers in ascending/descending order 8. to search a number or character from a string. 9. Moving Block Of Data From One Memory Location To Another Memory Location 10.to rotate the Stepper Motor in clockwise as well as anti-clockwise direction. **Outcomes**: result of arithmetic operations **Theoretical Background:** 8086 instruction set, MASM **Experimentation:** using MASM 1. Create Source File : Edit filename. Asm 2. process the source file with an assembler : Masm filename. asm X, Z 3. process the object file with linker: LINK filename. Obj linker produces a link file with the .EXE extension 4. execute .EXE: filename **Results and Discussions:** All programs are performed using 8086 instructions. **Conclusion:**	**12 Hrs.**
Textbooks: 1. Fundamental of Digital Circuits –A. Anand Kumar, 2nd Edition, PHI Private Limited. 2. Douglas Hall, "Microprocessors & Interfacing", McGraw Hill, Revised 2nd Edition, 2006 ISBN 0-07-100462-9 3. A.Ray, K.Bhurchandi, "Advanced Microprocessors and peripherals: Arch, Programming & Interfacing",Tata McGraw Hill,2004 ISBN 0-07-463841-6	
References: 1]R. Jain, "Modern Digital Electronics", eighth edition, Tata McGraw-Hill, 2003, ISBN 0 – 07 – 049492 – 4 2] John Uffenbeck," The 8086/88 Family: Design, Programming & Interfacing", PHI, 3] Liu, Gibson, "Microcomputer Systems: The 8086/88 Family", 2nd Edition, PHI,2005	
Experiment wise Measurable students Learning Outcomes: 1. understand all logic gates with its truth table 2. Study combinational circuit using DeMorgan's theorem 3. Design circuits using various logic gates 4. Understand code conversion using circuits. 5.Setup different circuits using kits like counters, flip-flops etc 6. Write assembly language program using instruction set.	

Title of the Course: Data Structures Lab Course Code:UITE0333	L -	T -	P 4	Credit 2

Course Pre-Requisite: Fundamentals of Programming Language C, and various data structures

Course Description:

The objective of this lab is to teach students various data structures and to explain them algorithms for performing various operations on these data structures. Students will gain practical knowledge by writing and executing programs in C using various data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.

Course Objectives:

1. To study the representation, implementation and applications of linear and non linear data structures.
2. Compute the complexity of various algorithms.
3. To develop application using data structure algorithms.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	Demonstrate the different data structures.	2	Understanding
CO2	Identify and make use of the appropriate data structure and algorithm to solve real world applications.	3	Applying
CO3	Compare different implementations of data structures and recognize the advantages and disadvantages of them.	4	Analyzing
CO4	Develop the real time application using data structure.	5	Create

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	PSO1	PSO 2
CO 1	2												1	
CO 2		3	2	2	1									3
CO 3		2		3									2	
CO 4	1	2	3	2	1	1			3	2	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 oral examination	
Course Contents:	
Experiment No. 1: Basic C programs **Aim and Objectives:** (I)Write a program to 1. find factorial of a given no. 2. determine the sum and average of a series. Etc (II) Write a program to perform following sparse matrix operations 3. Addition 2. subtraction 3. Multiplication 4. Transpose (III)Write a program to perform following operation on file: 1.Read 2.Write 3.Update **Outcomes:** **Theoretical Background:** basics of C language **Experimentation:** Write and execute the program using gcc. **Results and Discussions:** All programs are performed. **Conclusion:**	**06Hrs.**
Experiment No. 2: Linked List **Aim and Objectives:** Write a program to create singly/circular/doubly linked list and perform following operations on them. 1. Insert 2. Delete 3. Search 4. Traverse **Outcomes:** **Theoretical Background:** Basics of linked list and algorithms of all operations on linked list **Experimentation:** Write and execute the program using gcc. 1.Create a node using structure 2.Dynamically allocate memory to node 3.Create and add nodes to linked list 4. Delete the nodes from linked list 5. Display all the nodes of linked list **Results and Discussions:** All programs are performed. **Conclusion:**	**12 Hrs.**
Experiment No. 3: Stack **Aim and Objectives:** 1. Write a program to create stack and perform following operations on stack using array and linked list. 1. Push 2. Pop 3. Traverse 2. To convert a given infix expression into its postfix Equivalent, Implement the stack using an array. **Outcomes:** **Theoretical Background:** Basics of stack and algorithms of all operations on stack **Experimentation:** Write and execute the program using gcc. 1. 1. Create Stack 2. Perform all the operation of stack using arrays/linked list 3. Display the content of stack at last. 2.	**06 Hrs.**

1. Create a stack 2. Read an infix expression 3. convert infix exression into postfix expression **Results and Discussions:** All programs are performed. **Conclusion:**	
Experiment No. 4:Queue **Aim and Objectives:** Write a program to create queue and perform following operations on queue using array and linked list. 1. Enqueue 2. Dequeue 3. Traverse **Outcomes:** **Theoretical Background:** Basics of queue and algorithms of all operations on queue **Experimentation:** Write and execute the program using gcc. 1. Create a Queue 2. Perform all the operation of queue using arrays/linked list 3. Display the content of queue at last. **Results and Discussions:** All programs are performed. **Conclusion:**	**06 Hrs.**
Experiment No. 5: Tree **Aim and Objectives:** Write a program to create BST of integers/ characters and perform following operations on BST. 1. Insert node 2. Traverse 3. find no.of leaf nodes 4. find no. of internal nodes 5. find height of tree **Outcomes:** **Theoretical Background:** Basics of BST, different traversing techniques, and algorthms **Experimentation:** Write and execute the program using gcc. 1. Read integers 2. Create binary tree with the property binary search tree 3. Visit the tree in inorder, preorder, postorder 4. Display the visited nodes 5. display no.of leaf nodes 6. display no. of internal nodes 7. display height of tree **Results and Discussions:** All programs are performed. **Conclusion:**	**06 Hrs.**
Experiment No. 6: Graph **Aim and Objectives:** Write a program to 1. create graph using adjacency matrix 2. Traverse graph using a. BFS b. DFS **Outcomes:** **Theoretical Background:** Basics of Graph, different traversing techniques, and algorthms **Experimentation:** Write and execute the program using gcc. 1. Take the graph as a input 2. Start at some vertex and traverse it using DFS/BFS 3. Apply the above procedure for all nodes **Results and Discussions:** All programs are performed.	**04 Hrs.**

Conclusion:	
Experiment No. 7: Searching Techniques **Aim and Objectives:** Write a program for 1. Linear Search 2. Binary Search **Outcomes:** **Theoretical Background:** Linear and binary search algorithms **Experimentation:** Write and execute the program using gcc. **Linear Search** 1. Read the array 2. Search the element 3. Display the element position if found **Binary Search** 1. Read the sorted array 2. Search the element 3. Display the element position if found **Results and Discussions:** All programs are performed. **Conclusion:**	**04 Hrs.**
Experiment No. 8: Sorting Techniques **Aim and Objectives:** Write a program for 1. Bubble Sort 2. Selection Sort 3. Insertion Sort 4. Radix Sort 5. Quick Sort **Outcomes:** **Theoretical Background:** All sorting algorithms **Experimentation:** Write and execute the program using gcc. **Quick sort** 1. Read the elements to be sort 2. Find the proper pivot element 3. Apply quick sort method to sort the remaining elements **Selection sort** 1. Read the elements to be sort 2. Select the minimum element 3. Apply the selection sort to sort the remaining elements **Insertion Sort** 1. Start with an empty left hand [sorted array] and the cards face down on the table [unsorted array]. 2. Then remove one card [key] at a time from the table [unsorted array], and insert it into the correct position in the left hand [sorted array]. 3. To find the correct position for the card, we compare it with each of the cards already in the hand, from right to left. **Results and Discussions:** All programs are performed. Conclusion:	**10 Hrs.**
Experiment No. 9: Hashing **Aim and Objectives:** Write a program to create Hash Table and 1.Insert the data in hash table using hash function	**02 Hrs.**

2. Search the data in hash table
3. display the data of hash table

Outcomes:

Theoretical Background: Hashing, Hash table, Hash functions

Experimentation: Write and execute the program using gcc.

1. Read the key elements from the user
2. Use the hash function to store data in hash table
3. Apply required operation on the hash table

Results and Discussions: All programs are performed.

Conclusion:

Textbooks:

1. R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning, ISBN 9788131503140(Refer 1,3,4,5).
2.Seymour Lipschutz,G.A.V.Pai ,"Data Structure",Tata McGraw Hill,ISBN-13:978-0-07-060168-0.(Refer Chapter:6)
3.Rohit Khurana,"Data Structures Using C",Vikas publishing House Pvt.ltd,ISBN:978-93259-7565-1.(Refer Chapter:2 and application of DS)

References:

1.E. Horowitz , S.Sahani, S.Anderson-Freed "Fundamentals of Data Structures in C", Universities Press ,2008 ,ISBN 10:8173716056.

Experiment wise Measurable students Learning Outcomes:

1. Revise the C concepts
2. Implement array data structure to solve various problems
3. Implement linked list data structure to solve various problems
4. Solve problems using stacks data structures
5. Solve problems using queue data structures
6. Apply tree data structure to solve various problems
7. Implement different graph traversal techniques
8. Implement various kinds of searching techniques
9. Assess how the choice of data structures and algorithm design methods that impacts the performance of programs.
10. Solves problems using hashing techniques

Title of the Course: Automata Theory Course Code:UITE0401	L 3	T 1	P -	Credit 4

Course Pre-Requisite: Discrete Mathematics

Course Description: Automata theory is a subject mainly deals with verification and validation of event-action based processes in wide variety of engineering fields.

Course Objectives:

1. To interpret the formal languages like Regular Language and Context free Language.
2. To interpret the representation of Regular language as Regular Expression and Context free languages as context free grammar
3. To construct Language acceptors like finite automata for Regular Language and Push Down Automata for Context free Language.
4. To interpret properties of Context free languages in parsing.
5. To contrast the Turing Machines and its types.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	summarizes the formal languages like Regular Language and Context free Language	L2	Summarize
CO2	Constructs Language acceptors like finite automata for Regular Language and Push down Automata for Context free Language.	L3	Construct
CO3	Analyse properties of Context free languages in parsing.	L4	Analyze
CO4	Compare types of the Turing Machines.	L2	Compare

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	2										1			
CO 2		2									1			1
CO 3			1								1			1
CO 4	2													
CO 5														

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 1: Introduction to languages Strings, languages, grammar, types of grammars and languages, Regular expressions and corresponding regular languages , examples and applications, unions, intersection & complements of regular languages, Applications of regular expressions	**06 Hrs.**
Unit 2: Finite Automata Finite automata definition and representation, Non deterministic F.A. , NFA with null transitions, Equivalence of FA's , NFA's and NFA's with null transitions. Equivalence of regular expressions and finite automata, Minimization of Finite Automata, Application of Finite Automata	**08 Hrs.**
Unit 3: Context Free Grammar Context Free Grammar- Definition and examples, regular grammar, languages of a grammar, BNF and CNF notations, derivation and parse tree, Ambiguity in grammars and languages: removal of ambiguity, Normal forms, converting CFG to CNF form, Application of Context free grammar	**06 Hrs.**
Unit 4: Push Down Automata Definition, The Language of PDA, Deterministic PDA and Non Deterministic PDA, Acceptance by Final state and empty stack, Equivalence of PDA's and CFG: CFG to PDA, PDA to CFG	**08 Hrs.**
Unit 5: Parsing Parsing – Top-Down, Recursive Descent and Bottom-Up Parsing, Pumping lemma for Context free language, intersection and complement of Context free language	**06 Hrs.**
Unit 6: Turing Machine Turing Machines- definition of TM as Language acceptors and examples, combining Turing machines, computing a partial function with a TM, Variations in TM- single tape, multitape, Non-deterministic TM and Universal TM, Church- Turing machine.	**08 Hrs.**

Textbooks:

1. Introduction to languages & Theory of computations – John C. Martin (MGH)

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)

Unit wise Measurable students Learning Outcomes:

Unit 1: Introduction to languages

UO 1.1:To interpret regular expressions and its applications
UO 1.2:To interpret the regular languages defining

Unit 2: Finite Automata

UO 2.1:To interpret the design method of FA.
UO 2.2:To summarize the mapping of regular expression accept by FA .

Unit 3: Context Free Grammar
UO 3.1:To interpret the regular grammar used in regular languages.
UO 3.2:To construct regular languages for various acceptance languages

Unit 4: Push Down Automata
UO 4.1:To summarize push down automata
UO 4.2: To design mapping of CFG to PDA and vice verse

Unit 5: Parsing
UO 5.1:To analyze the parsing in compilation.

Unit 6: Turing Machine
UO 6.1:To interpret turing machines model

Title of the Course: Software Engineering Course Code: UITE0402	L 3	T 1	P -	Credit 4

Course Pre-Requisite: Problem Solving Using C

Course Description: This course gives you fundamentals of software development in the current IT industry. The fundamentals are divided into different parts. The first part deals with different software models followed for development of software. The subsequent parts deals with requirement specification, software design with UML, coding and testing respectively. You will get complete insight of software development process which will help you a lot in your career in IT industry.

Course Objectives:

1. Explore different software process models such as the waterfall and evolutionary models
2. Describe different components of software requirement specification document
3. Elaborate different software level concepts wrt to software design approaches
4. Explain different coding standards used in software coding
5. Outline different types of testing methodologies

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	Compare SE approaches with different software development process models	IV	Analyzing
CO2	Develop different components of software requirement specification (SRS)	III	Applying
CO3	Design software using structured design methodology and object oriented methodology for any real world problem	VI	Creating
CO4	Develop coding for different software designs	III	Applying
CO5	Compare different types of testing techniques	II	Understanding

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**													
CO2	**3**													2
CO3					**2**									2
CO4					**2**									
CO5			**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 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: Introduction and Software Process The Problem Domain, SE Challenges, SE Approaches, Software Process, Desired Characteristics of a Software Process, Software Development Process Models- Waterfall Model, Prototype Methodology, Agile Software Development Methodology, Rapid Application Development (RAD),Dynamic Systems Development Model Methodology, Spiral Model, Extreme Programming Methodology, Feature Driven Development.	**10 Hrs.**
Unit 2 : Software Requirement Analysis and Specifications Software Requirements, Problem Analysis, Requirements Specification, Functional Specifications with use cases, Validation, Metrics	**06 Hrs.**
Unit 3 : Software Design Approaches Design Principles, Module-Level Concepts, Design Notation and Specification, Structured Design Methodology, OO Analysis and OO Design, OO Concepts, Design Concepts.	**07 Hrs.**
Unit 4 : UML Structural Modeling Classes,Relationship,Common Mechanics, Diagrams and Class Diagrams, Advanced Classes, Advanced Relationships, Interfaces, Types, and Roles, Packages, Instances and Object Diagram	**07 Hrs.**
Unit 5 : UML Behavioral and Architectural Modeling Behavioral: Interactions, Use Cases, Use Case Diagrams, Interaction Diagrams, Activity Diagrams Architectural: Components, Deployment, Collaborations, Patterns and Frameworks, Component Diagrams, Deployment Diagrams	**06 Hrs.**
Unit 6 : Coding and Testing Programming Principles and Guidelines, Coding Process, Refactoring, Verification, Metrics, Testing Fundamentals, Black-Box Testing, White-Box Testing.	**06 Hrs.**

Textbooks:

4. 'An Integrated approach to Software Engineering' –Pankaj Jalote, 3rd Edition, Narosa Publication. (1,2,3,6)

5. UML User Guide- Grady Booch, James Rumbaugh, Publisher: Addison Wesley (4,5)

References:

1) Software Engineering- A Practitioner's Approach – Roger S. Pressman (TMH)
2) Software Engineering- Ian Sommerville – Pearson
3) Software Engineering by Kogent Wiley India Limited.

Unit wise Measurable students Learning Outcomes:

Unit 1: Introduction

UO-1.1) Explain IT industry component

UO-1.2) Differentiate between program and software

UO-1.3) Enlist problems of software and software engineering

UO-1.4) Compare different software process models

UO-1.5) Describe project and configuration management process

Unit 2 : Software Requirement Analysis and Specification

UO-2.1) Describe software requirement process

UO-2.2) Design DFD using data flow modelling with example

UO-2.3) Design class diagram using OO modelling with example

UO-2.4) Develop components of SRS

UO-2.5) Apply metrics and validation for SRS

Unit 3: Software Design Approaches

UO-3.1) Apply design principles to real world problem

UO-3.2) Describe module level concepts of FOA and OOA

UO-3.3) Enlist design and notation used for design

UO-3.4) Apply design methodology to real world problem

Unit 4 : UML Structural Modeling

UO 4.1) Explain classes and relationships
UO 4.2) Design class diagram for real world problem
UO 4.3) Describe advanced class concepts

Unit 5: UML Behavioral and Architectural Modeling

UO 4.1) Explain concepts of interaction modeling
UO 4.2) Design interaction diagram for real world problem
UO 4.3) Explain concepts of architecture modeling
UO 4.4) Design architecture diagram for real world problem

Unit 6: Coding and Testing

UO-5.1) Explain programming principles and guidelines

UO-5.2) Describe coding process

UO-5.3) Apply refactoring with example UO-5.4) Enlist different types of testing. UO-5.5) Describe testing process

Title of the Course: Computer Networks	**L**	**T**	**P**	**Credit**
Course Code: UITE0403	**3**	**-**	**-**	3

Course Pre-Requisite: TCP/IP Protocol Suit

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

5. To make students able to use and analyze various Protocols using Protocol analyzing tools like wireshark and tcpdump.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		level	**Descriptor**
CO1	recall the basic terminology used in client/server pro-gramming	1	Remembering
CO2	describe architecture/working of various internet appli-cations/protocols.	2	Understanding
CO3	install and con gure FOSS server to provide services in internet.	3	Applying
CO4	analyze various protocols of TCP/IP protocol suite us-ing wireshark and tcpdump.	4	Analyzing
CO5	build complete application for use in Internet using Javaprogramming.	6	Creating

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	1									2				
CO2	2									2				
CO3	3					2			3		2			
CO4		3		3	2			2				2		
CO5			3					2						

Assessments :

Teacher Assessment:

Two components of In-Semester-Evaluation (ISE), One Mid-Semester Examination (MSE) and One End-Semester-Examination (ESE) will have 20%, 30% and 50% weights respec-tively.

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 with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: 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	**08 Hrs.**
Unit 2: Transport Layer Transport layer functions, UDP- datagram, services, applications, TCP - services, seg-ment, connection, state transition diagram, ow control, congestion control, error control, timers.	**08 Hrs.**
Unit 3: Introduction to Application Layer Client-Server paradigm, client, server, concurrency, socket interface, communication using using tcp, communication using udp.	**06 Hrs.**
Unit 4: DHCP, DNS, FTP and TFTP DHCP: Introduction, Previous Protocols, DHCP operation, Packet Format, DHCP Con-guration. DNS: Need, Name Space, Domain Name Space, Distribution of name space, and DNS in internet, Resolution, DNS massages, Types of records, Compression examples, encap-sulation. FTP: Connections, Communication, Command processing, File transfer, User interface, Anonymous FTP, TFTP.	**09 Hrs.**
Unit 5: HTTP, Electronic Mail, SNMP HTTP: Architecture, Web Documents, HTTP Transaction, Request & Response mes-sages: 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.	**09 Hrs.**
Unit 6: 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.	**08 Hrs.**

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

References:

Unit wise Measurable students Learning Outcomes:
At the end of each of following unit, student should be able to:

Unit 1: Network Layer

UO-1.1 To list advantages of IPv6.

UO-1.2 To compare IPv4 with IPv6 and ICMPv4 with ICMPv6.

UO-1.3 To summarize strategies for transition from IPv4 to IPv6.

Unit 2: Transport Layer

UO-2.1 To list services of Transport layer.

UO-2.2 To compare TCP and UDP services.

Unit 3: Introduction to Application Layer

UO-3.1 To recall the terminology used in client-server paradigm.

UO-3.2 To develop application for client-server communication over TCP/UDP.

UO-3.3 To analyze protocols using wireshark or tcpdump.

Unit 4: DHCP, DNS, FTP and TFTP

UO-4.1 To explain need and working of DHCP, DNS, FTP.

UO-4.2 To use services of DHCP, DNS, FTP.

UO-4.3 To capture and examine packets of DHCP, DNS, FTP.

UO-4.4 To understand and develop application using these protocols.

UO-4.5 To demonstrate use of di erent user agents to access services of DHCP, DNS, FTP, TFTP.

Unit 5: HTTP, Electronic Mail, SNMP

UO-5.1 To explain the working of WWW.

UO-5.2 To explain the mechanism of sending and receiving e-mails.

UO-5.3 To develop applications using SMTP and POP3.

UO-5.4 To tell the use of SNMP.

Unit 6: Multimedia in Internet

UO-6.1 To explain use of Internet for audio/video interaction.

UO-6.2 To list applications and associated protocols for di erent a/v interactions.

UO-6.3 To discuss the working of protocols RTP and VoIP.

Title of the Course: Computer Organization and Architecture Course Code:UITE0404	L 3	T -	P -	Credit 3

Course Pre-Requisite: Fundamentals of Programming Languages-I & II and Basics of Electronics Engineering

Course description: To impart the necessary fundamental principles that is essential to study courses in computer science and related fields. To develop hardware knowledge necessary for IT engineer.

Course Objectives:

1. To understand the structure, function and characteristics of computer systems.
2. To identify the elements of modern instructions sets and explain their impact on processor design.
3. To explain the function of each element of a memory hierarchy, identify and compare different methods for computer I/O.
4. To compare simple computer architectures and organizations based on established performance metrics.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	**Interpret** the history and development of modern computer	II	Understanding
CO2	**Select** various data representations and explain how arithmetic and logical operations are performed by computers	III	Applying
CO3	**Explain** organization of digital computers and explain the basic principles and operations of different components	II	Understanding
CO4	**Evaluate** the performance of CPU	V	Evaluating
CO5	**Distinguish** the representation of data, addressing modes, instructions sets		
CO6	**Explain** the concepts of parallel processing	II	Understanding

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PSO1	PSO2
CO1	1			2		1						2	
CO2				3									1
CO3				3								2	
CO4	2	2		2								2	
CO5	1	2										2	
CO6	1	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 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: Computer Evolution and Performance Computer Organization and Architecture, Structure and Function, Evolution (a brief history) of computers, Designing for Performance, Evolution of Intel processor architecture- 4 bit to 64 bit, performance assessment. A top level view of Computer function and interconnection- Computer components, Computer Function, Interconnection structure, bus interconnection	**6 Hrs.**
Unit 2: CPU Organization Fundamentals, Study of design and architecture of a small accumulator based CPU, Architecture extensions, a typical CPU with general register organization, pipelining, RISC Machines: Organization of ARM6, CISC Machines: Organization of 68020 Data representation: Fixed- Point Numbers, Floating Point Number- The IEEE 754 floating pointing numbers, Instruction Set: Instruction Formats, Addressing Modes, Self-Study: Instruction Types.	**7 Hrs.**
Unit 3: Data path Design Addition-Subtraction-High speed adders-- A 4-bit carry-lookahead adder, Design of a complete twos-complement adder- subtractor, Multiplication: twos-complement multiplier, Robertson multiplication algorithm for twos–complement fractions, Booths multiplication algorithm, Division: Non-restoring division algorithm for unsigned integers.	**7 Hrs.**
Unit 4: Control Design Hardwired Control: Design of DMA controller, Design Examples: Multiplier Control, Implementing a multiplier control unit, CPU control unit: Control unit design: Implementing a program control unit.	**8 Hrs.**
Unit 5: Memory Organization Memory: Memory device Characteristics, Random access memories: A commercial 64Mb DRAM chip, Serial-Access Memories: A commercial magnetic hard-disk memory unit, Memory Systems: Multilevel memories, Address translation, Memory allocation, Caches: Cache organization, Cache operation, Address Mapping.	**7 Hrs.**
Unit 6: Parallel Processing	**7 Hrs.**

Parallel Computer Structures: Pipeline Computers, Array computers, Multiprocessor systems, performance of parallel computers, Dataflow and New concepts; Architectural Classification Schemes: Multiplicity of Instruction- Data streams, Serial vs Parallel processing, Parallelism v/s pipelining	

Textbooks:

1. John P Hays, ― Computer Architecture and Organization‖, McGraw-Hill Publication, 1998, ISBN:978-1-25-902856-4, 3rd Edition.
2.W. Stallings, ― Computer Organization and Architecture: Designing for performance‖, Pearson Education/ Prentice Hall of India, 2003, ISBN 978-93-325-1870-4, 7th Edition.
3.Zaky S, Hamacher, ― Computer Organization‖, 5th Edition, McGraw-Hill Publications, 2001, ISBN- 978-1-25-900537-5, 5th Edition.

References:

1] A. Tanenbaum, ― Structured Computer Organization‖, Prentice Hall of India, 1991 ISBN: 81 – 203 – 1553 – 7, 4th Edition
3] Patterson and Hennessy, ― Computer Organization and Design‖, Morgan Kaufmann Publishers In, ISBN 978-0-12-374750-1, 4th Edition.

Unit wise Measurable students Learning Outcomes:

Unit 1: Computer Evolution and Performance

ULO 1.1 Understand basics of computer organization

ULO1.2 Understand the structure, function and characteristics of components of computer and computer systems

Unit 2: CPU Organization

ULO 2.1 Study design and architecture of different CPU's , Architecture and extensions

ULO 2.2 Study different representation of data types.

Unit 3: Data path Design

ULO 3.1 Design the component of computer, like data path unit

ULO 3.2 Study design of adder-subtractor, multipliplication and division algorithms.

Unit 4: Control Design

ULO 4.1 Study design of control unit

Unit 5: Memory Organization

ULO 5.1 Study memory device characteristics, Random access memories.

ULO 5.2 Study addresses translation techniques.

Unit 6: Parallel Processing

ULO 6.1 Summarize the parallel processing concepts

Title of the Course: Object Oriented Programming Course Code:UITE0405	L	T	P	Credit
	3	-	-	3

Course Pre-Requisite: Computer programming in C

Problem solving techniques

Course Description: This course aims to study the object-oriented programming principles and techniques'++ under Linux platform is used as programming language.

Course Learning Objectives [CLO]:

CLO-1: To introduce students with Object Oriented Programming (OOP) paradigm.
CLO-2: To make students familiar with C++ programming language.

CLO-3: To make students able to write applications using OOP concepts.

CLO-4: To make students able to use built-in classes from STL.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Bloom's Cognitive Descriptor
CO1	explain object oriented concepts, principles and techniques.	2	Understanding
CO2	apply various object oriented features to solve various computing problems using C++ language.	3	Applying
CO3	apply exception handling and use built-in classes from STL	3	Applying
CO4	compare procedure oriented programming and object oriented programming	4	Analysing
CO5	build a complete application using bottom-up design approach with team-work in mind	6	Creating

CO-PO Mapping:

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

5														

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 with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction

Introduction to procedural & object-oriented programming, Limitations of procedural programming, Need of object-oriented programming, 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

Variable declarations, global scope, const variables, reference variables, function proto-types, functions with default arguments, call by value, call by reference, returning by reference, call by pointer, inline functions, constant arguments, `cin', `cout', formatting and I/O manipulators, Classes and Objects defining Class, data members, member functions, Access specifiers, public, private, protected, constructor, destructor, array of objects, passing objects to functions, returning object.

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

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 over loading and operator overloading, opera-tor 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

Unit 5: Files and Streams

Concept of Streams, concept of File, opening and closing a le, detecting end-of- le, le modes, le pointer, reading and writing characters, strings and objects to the le, operations to move le pointers i.e seekg, seekp, tellg, tellp.

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

Text Books:

- C++: The Complete Reference Fourth Edition - Herbert Schildt (McGraw-Hill)
- C++ programming: From Problem Analysis to Program Design Fifth Edition -D.S. Malik (Cengage Learning)
- C++ Programming with language –Bjarne Stroustrup (AT & T)

Reference Books:

- Object Oriented Programming with C++ Fourth Edition-E Balguruswamy (McGraw Hill)
- Object oriented Programming in C++ 3rd Edition-R.Lafore (Galgotia Publications)
- C++ programming –John Thomas Berry(PHI)
- Object –Oriented Analysis & Design: Understanding System Development with UML 2.0 , Docherty, Wiley India Ltd.
- http://www.spoken-tutorial.org/ NMEICT Project of Govt. Of India

Unit wise Measurable students Learning Outcomes:

Unit 1: Introduction

UO1-1: Identify Basic difference between c & c++.
UO1-2: Identify Basics OOP.

Unit 2: Basics of C++ programming

UO2-1: Explain use of cout and cin for input output operations.
UO2-2: Implement class with proper use of access specifiers.

Unit 3: Inheritance

UO3-1: Define parent child relationship between classes.
UO3-2: Understand various forms and types of inheritance.

Unit 4: Polymorphism

UO4-1: define polymorphism and method overloading and overriding.
UO4-2: Apply concept of operator overloading.

Unit 5: Files & Streams

UO5-1: Implement different file handling programs.

Unit 6:Advace C++ features

UO6-1: Implement programs using Exception handling
UO6-2: Solve problems using STL

Title of the Course: Object Oriented Programming Lab Course Code:UITE0431	L	T	P	Credit
	-	-	2	1

Course Pre-Requisite: Fundamentals of Programming Language C

Course Description:
This course aims to study the object-oriented programming principles and techniques'++ under Linux platform is used as programming language.

Course Objectives:
CLO-1: To introduce students with Object Oriented Programming (OOP) paradigm.
CLO-2: To make students familiar with C++ programming language.

CLO-3: To make students able to write applications using OOP concepts.

CLO-4: To make students able to use built-in classes from STL.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	apply various object oriented features to solve various computing problems using C++ language.	3	Applying
CO2	apply exception handling and use built-in classes from STL	3	Applying
CO3	compare procedure oriented programming and object oriented programming	4	Analyzing
CO4	build a complete application using bottom-up design approach with team-work in mind	6	Creating

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	PSO1	PSO 2
CO 1	3	2		2	3								3	
CO 2		3	2	3	3		2						3	2
CO 3						2	3	3		3	1			3
CO 4							2	2	3	2		3	2	3

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 oral examination

Course Contents:

Experiment No. 1 **Aim and Objectives:** (I) Implementation of Inline functions, functions with default arguments, reference parameters **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**04 Hrs.**
Experiment No. 2 **Aim and Objectives:** 2. Implementation of Class Objects, Constructor, destructor, constructor overloading **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**10 Hrs.**
Experiment No. 3: **Aim and Objectives:** Implementation of Operator and function overloading **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**04 Hrs.**
Experiment No. 4: Implementation of Multiple and multilevel inheritance using virtual base class **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**04 Hrs.**
Experiment No. 5: .Demonstration of Pointers- new, delete operators **Outcomes:** **Theoretical Background:** Basics of BST, different traversing techniques, and algorthms **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++.	**06 Hrs.**

Results and Discussions: All programs are performed.. **Conclusion:**	
Experiment No. 6: Implementation of Friend function, friend class **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**04 Hrs.**
Experiment No. 7: Implementation of class and function Templates **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**04 Hrs.**
Experiment No. 8: Implementation of Exception Handling **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. Conclusion:	**10 Hrs.**
Experiment No. 9: Implementation of File Handling and STL using OOP concepts **Outcomes:** **Theoretical Background:** basics of C++ language **Experimentation:** Write and execute the program using g++. **Results and Discussions:** All programs are performed. **Conclusion:**	**02 Hrs.**

Text Books:

- C++: The Complete Reference Fourth Edition - Herbert Schildt (McGraw-Hill)
- C++ programming: From Problem Analysis to Program Design Fifth Edition -D.S. Malik (Cengage Learning)
- C++ Programming with language –Bjarne Stroustrup (AT & T)

References:

- Object Oriented Programming with C++ Fourth Edition-E Balguruswamy (McGraw Hill)
- Object oriented Programming in C++ 3rd Edition-R.Lafore (Galgotia Publications)
- C++ programming –John Thomas Berry(PHI)

Title of the Course: Problem Solving using Computer	L	T	P	Credit
Course Code: UITE0432	1	-	2	2

Course Prerequisite: Fundamentals of Programming

Course Description:

This course aims at giving students a basic knowledge of Python Programming.

Course Objectives:

1. To develop problem solving skills.
2. To understand Python programming environment.
3. To write, run, document programs in python.

Course Learning Outcomes:

CLO	After the completion of the course the student should be able to	Bloom's Cognitive level	Bloom's Cognitive Descriptor
CLO1	Recall the basics concept of Python programming.	1	Remember
CLO2	Summarize different decision control and looping constructs in python	2	Understand
CLO3	Write python programs related to simple-moderate mathematical/logical problems.	3	Apply
CLO4	Experiment with concept of files handling and exception handling.	3	Apply

CO-PO Mapping:

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

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 oral examination

Course Contents:

Unit 1: Introduction to Python Features of Python, Identifiers, variables, comments, operators, Input/Output, Setting up python programming environment.	**02 Hrs.**
Unit 2: Python Data Structures Boolean, Numbers, Strings, List, Tuple, Dictionary, Set	**02 Hrs**
Unit 3: Control Flow and Function **Control Flow:** Decision Making, Control Statements, Loops, Nested loops **Function:** Function definition, calling function, function argument, variable scope, recursive function	**02 Hrs.**
Unit 4: Modules and Packages What are Modules?, Built in Modules, Creating modules, import statement, Namespace and scope, Packages	**02 Hrs.**
Unit 5: File Handling **File Object:**File Built-in function, File Built-in methods, File Built-in Attributes, standard files and command line arguments.	**02 Hrs.**
Unit 6: Exception Handling Built-in Exception, Detecting and Handling exception-*try...except, try* statement with multiple *except, except* statement with multiple exception, catching all exception, Exceptional arguments, *else* clause, *Finally* clause, *try-finally* statement	**02 Hrs.**

Textbooks:

- Wesley J Chun,"Core Python Programming", Second Edition, Pearson Publication

References:

1. Jeeva Jose & P. Sojan Lal, "Introduction to computing and Problem Solving with Python", Khanna Book Publishing Co. (p) Ltd
2. Timothy A. Budd, "Exploring Python", Tata McGraw-Hill Publication

Unit wise Measurable students Learning Outcomes:

Unit 1:Introduction to Python

UO 1.1: Describe the Identifiers, variables, comments, operators, Input/Output,

Unit 2: Python Data Structures

UO 2.1: Explain Boolean, Numbers, Strings, List, Tuple, Dictionary, Set.

Unit 3: Control Flow and Function

UO 3.1: Explain different Decision Making statements and Functions.

Unit 4: Modules and Packages

UO 4.1: Create and import modules and packages.

Unit 5: File Handling

UO 5.1: Understand and summarize different File handling operations.

Unit 6: Exception Handling

ULO 6.1: Experiment with exception Handling.

Course Contents:	
Experiment No. 1: Installation of Python **Aim and Objectives:** To download and install python on Windows/Linux. **Outcomes:** Student will be able to run python on Windows/Linux. **Theoretical Background:** **Experimentation:** **Results and Discussions:** **Conclusion:**	**02 Hrs.**
Experiment No. 2: **Aim and Objectives:** To understand basic concept of Identifiers,reserved keywords,variables,comments,operators,Numbers. **Outcomes:** **Theoretical Background:** Identifiers,reserved keywords, variables, comments, operators,Numbers. **Experimentation:** **1.** Write a program that declares 3 integers, determines and prints the largest and smallest in the group. **2.** Write a program that accepts two numbers from the user and print their sum. **3.** Write a program to calculate simple interest. **Results and Discussions:** All programs are performed. **Conclusion:**	**02 Hrs.**
Experiment No. 3: **Aim and Objectives:** To understand basic concept String,List and Tuples,set,dictionaries **Outcomes:** **Theoretical Background:** String,List and Tuples,set,dictionaries **Experimentation:** 1. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged. Sample String : 'abc' Expected Result : 'abcing' Sample String : 'string' Expected Result : 'stringly' 2. Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples. Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)] Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)] 3. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).	**02 Hrs.**

SampleDictionary(n=5) : Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25} **4.** Write a Python program to create an intersection,union,difference of sets. **Results and Discussions:** All programs are performed **Conclusion:**	
Experiment No. 4: Decision Making **Aim and Objectives:** To understand different decision making statements. **Outcomes:** **Theoretical Background: Decision Making-if,if..else,if..elif...else** **Experimentation:** 1.Write a program to find GCD of two numbers. 2.Write a python program to check wheather a specified value is contain in list. 2.Write a program that prints the numbers from 1 to 20. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz". **Results and Discussions:** All programs are performed **Conclusion:**	**02 Hrs.**
Experiment No. 6: Function **Aim and Objectives:** To understand concept of function. **Outcomes:** **Theoretical Background:** function. **Experimentation:** 1.Write a program to find factorial of a number using recursion. 2.Write a program to find LCM of two number with and without using GCD 3. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters. **Results and Discussions:** All programs are performed **Conclusion:**	**02 Hrs.**
Experiment No. 7: Modules and Packages **Aim and Objectives:** To understand concept of Modules and Packages. **Outcomes:** **Theoretical Background:** **Experimentation:** 1. Write a program to shuffle a deck of cards using random module. 2. Write a program to check whether given string is palindrome or not. Import the package to see whether input string is palindrome or not. 3. Write a python function that print out the first n rows of Pascal Triangle. Import the module to accept n from user. **Results and Discussions:** All programs are performed **Conclusion:**	**02 Hrs.**
Experiment No. 8: File Handling **Aim and Objectives:** To understand concept of File handling. **Outcomes:** **Theoretical Background:**	**02 Hrs.**

Experimentation: 1. Write a program to append a file with the content of another file. 2. Write a program to insert a sentence and delete a sentence from the specified position from the file. 3. Write a python program to search a word and replace with another word for all the occurrences in file. **Results and Discussions:** All programs are performed **Conclusion:**	
Experiment No. 9: Exception Handling **Aim and Objectives:** To understand concept of Exception Handling. **Outcomes:** **Theoretical Background:** **Experimentation:** 1. Write a function to compute 5/0 and use try/except to catch the exceptions. 2. Write a program to handle exception for attempting to open a non-existent file. **Results and Discussions:** All programs are performed	**02 Hrs.**

Course Name: **Problem Solving Using Computers (Lab)**

Course Code: **UITE0432**

Problem Statements:

PSUCPBLPB01:Implement Dice Rolling Simulator

"The Goal: Like the title suggests, this project involves writing a program that simulates rolling dice. When the program runs, it will randomly choose a number between 1 and 6. (Or whatever other integer you prefer — the number of sides on the die is up to you.) The program will print what that number is. It should then ask you if you'd like to roll again. For this project, you'll need to set the min and max number that your dice can produce. For the average die, that means a minimum of 1 and a maximum of 6. You'll also want a function that randomly grabs a number within that range and prints it."

PSUCPBLPB02: Implement method to guess the Number.

"This project also uses the random module in Python. The program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to *input* information.) If the user's guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear. You'll need functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers."

PSUCPBLPB03: Implement Mad libs Generator

"The program will first prompt the user for a series of inputs a la Mad Libs. For example, a singular noun, an adjective, etc. Then, once all the information has been inputted, the program will take that data and place them into a premade story template. You'll need prompts for user input, and to then print out the full story at the end with the input included.**"**

PSUCPBLPB04: Implementation of TextBased Adventure Game

"A complete text game, the program will let users move through rooms based on user input and get descriptions of each room. To create this, you'll need to establish the directions in which the user can move, a way to track how far the user has moved (and therefore which room he/she is in), and to print out a description. You'll also need to set limits for how far the user can move. In other words, create "walls" around the rooms that tell the user, "You can't move further in this direction."

PSUCPBLPB05: Implementation of Hangman

"The main goal here is to create a sort of "guess the word" game. The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you'll need a way to grab a word to use for guessing. (This can be grabbed from a pre-made list. No need to get too fancy.) You will also need functions to check if the user has actually inputted a single letter, to check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses."

PSUCPBLPB06: Implement the turtle race game

"In this project you will use loops to create a racing turtle game and draw a race track. Need to create vertical race tracks. The tracks should be drawn fast. Now for the fun bit add some racing turtles. It would be really boring if the turtles did the same thing every time so they will move a random number of steps each turn. The winner is the turtle that gets the furthest in 100 turns. "

PSUCPBLPB07:Create a dictionary of colors

"In this project you will create a dictionary of colors which maps hard to remember color codes into friendly names. hex color codes is really flexible but they are hard to

remember. In Python, a dictionary is even more flexible so create a dictionary to map from human-friendly color names (keys) to computer-friendly hex codes (values)."

2. Activities with timeline:

Sr. No.	Activity	Timeline
1	PBL awareness in class	1st week
2	Announcement of problem/s for PBL	2nd week
3	Team formation	3rd week
4	Project ISE I:Synopsis presentation	5th week
5	Completion of corrections/improvements in synopsis	6th week
6	Project ISE II: Project Progress Presentation with Model/case study	10th week
7	Completion of correction/improvements in Evaluation II	11th week
8	End Semester Evaluation of Project	13th week
9	Determining future scope for improvement	14th week

3. Assessment Scheme:

- ISE-I
- ISE-II
- Project ESE

4. Evaluation Scheme:

- **Project ISE I :** Synopsis presentation for 5 marks (evaluation with rubrics)
- **Project ISE II:** Project Progress Presentation with Model/case study for 5 marks (evaluation with rubric)
- **End Semester Evaluation of Project:** Multimedia presentation and demonstration of working models for 15 marks out of 25 of course lab ISE (evaluation with rubrics)

Title of the Course: Computer Networks Lab Course Code:UITE0433	L	T	P	Credit
	-	-	2	1

Course Prerequisite: Fundamental knowledge C Programming Language.

Course Description:

The objective of this lab course is to provide students solid understanding of the various protocols of TCP/IP protocol suite. Students will gain practical knowledge by writing and executing programs in C/C++ using socket programming. Students will be also able to analyze standard internet protocols.

Course Objectives:

After completion of this course students will be able to

1. identify client-server model and implement it using socket programming.
2. understand working of various Internet services.
3. use protocol analysing tools.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1	write applications using TCP/IP protocols	3	Applying
CO2	install and configure FOSS server to provide services in internet.	3	Applying
CO3	analyze various protocols of TCP/IP protocol suite.	4	Analyzing

CO-PO Mapping:

CO	PO1	PO 2	PO3	PO 4	PO5	PO 6	PO 7	PO8	PO 9	PO10	PO 11	PO 12	PSO1	PSO2
CO1		2												
CO2		3	2	2	1									3
CO3		2		3									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 Oral Examination

Course Contents:

Experiment No. 1: Study socket programming API in C/C++.	02 Hrs

Aim and Objectives: To study APIs of C/C++ used for socket programming: socket(), bind(), listen(), connect(), accept(), read(), write(), sendto(), recvfrom() To study the steps required at server and client side of application.	
Experiment No. 2: Well known client **Aim and Objectives:** Write client application to get the services of simple well known client using TCP/UDP like daytime, echo, finger, etc.	**02 Hrs.**
Experiment No. 3: Iterative UDP Server / Client **Aim and Objectives:** Write application using UDP to provide iterative service.	**02 Hrs**
Experiment No. 4: Concurrent TCP Server / Client **Aim and Objectives:** Write application using TCP to provide concurrent service.	**02 Hrs**
Experiment No. 5: Communicating using IPv6 **Aim and Objectives:** Write application using TCP/UDP to provide communication using IPv6.	**02 Hrs**
Experiment No. 6: Packet Capturing and analysis **Aim and Objectives:** Study wireshark/ethereal to capture packets in network and do analysis of different protocols.	**02 Hrs**
Experiment No. 7: Web client **Aim and Objectives:** Write application to fetch web page from web server using HTTP messages.	**02 Hrs.**
Experiment No. 8: SMTP user agent **Aim and Objectives:** Write program to send email using SMTP commands.	**02 Hrs.**
Experiment No. 9: POP3 user agent **Aim and Objectives:** Write program to send retrieve email using POP3 commands.	**02 Hrs.**
Experiment No. 10: FOSS Server **Aim and Objectives:** Install any FOSS server. Configure it to provide service in LAN and Internet.	**02 Hrs.**

Title of the Course : Mini Project-I	L	T	P	Credit
Course Code : UITE0441	-	-	2	1

Course Pre-Requisite: C,C++ or any free and open source software

Course Description: This course aims at developing mini project based on technologies learnt.

Course Objectives:

Students should be able

1. To apply Software engineering approach to solve real life problem.
2. To learn skills of team work & team building to accomplish common goal
3. To design and develop logical skills to use appropriate data structures for solving real life engineering problem.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO 1	To apply software engineering approach to solve real life problem.	2 & 3	Understanding& Applying
CO 2	Learn the skills of team work & team building to accomplish common goal	2	Understanding
CO 3	Design and Develop logical skills to use appropriate data structures for solving real life engineering problems.	5	Creating

CO-PO Mapping:

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO 11	PO 12	PSO 1	PSO2
CO 1	3				1	2			3	1			1	
CO 2		3	2			1			3	3			1	3
CO 3			3			2			2	2	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 oral examination
Description: The mini project should be undertaken preferably by a group of 3-4 students who will jointly work and implement the project. The mini project must be based upon any real life problem statement. Platforms: Free and Open source software. 8. The group will select a problem with the approval of the guide and prepare the solution guidelines for its implementation. 9. The same should be put in the form of synopsis (3 to 5 pages), stating the usage of logic, algorithms and suitable data structures necessary for implementation of the solution as per software engineering approach. 10. Further the group is expected to complete analysis of problem by examining the possible different inputs to the system and the corresponding outputs. 11. The term work submission is to be done in the form of a report containing the details of the problem, solution techniques, implementation details, input-output scenarios and the conclusion. The project must be implemented using technologies covered earlier in previous semester.

Title of the Course: Soft Skills	L	T	P	Credit
Course Code:UITE0461	1	1	-	-

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	Bloom's Cognitive level	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	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1												1	1	
CO2												1	1	
CO3								2					2	
CO4												1	1	
CO5									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.

Unit 6: Job Interview:

UO6.1) To explain the importance of job interview techniques.
UO6.2) To make use of job interview skills while facing an interview.