KIT's College of Engineering (Autonomous), Kolhapur

Department of Computer Science & Engineering

S. Y. CSE - Syllabus

Computer Science & Engineering

AY-2022-23

SEM-I

		L	T	P	Credit
Course Code:	UCSC0301				
Course Name:	Computational Mathematics	**3**	**1**		**4**

Course Prerequsites:

Statistics, Probability, Vectors and Set Theory

Course Description:

This Course contains Statistics, Probability, Vectors and Fuzzy Set

Course Outcomes:

CO1	After the completion of the course the student will be able to, Demonstrate the basic mathematical concepts in Computer Science Engineering related to Fuzzy Sets and Statistics
CO2	Explain Vector Space concepts in dealing with problems in Computer Science Engineering.
CO3	Apply the knowledge of Statistics to solve problems arising in Computer Science Engineering.
CO4	Apply the knowledge of Fuzzy Equation to solve problems arising in Computer Science Engineering.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination (MSE)	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1	**Advanced Linear algebra**	**6 Hours**

1.1 Solutions of simultaneous linear equations using Gauss-Jordan method.
1.2 Solutions of simultaneous linear equations using LU decomposition method.
1.3 Determination of Eigen Value by Iteration method.
1.4 Solution of non-linear simultaneous equations.

Unit 2	**Vector Algebra**	**8 Hours**

2.1 Vector Spaces,
2.2 Subspaces, basis, span,
2.3 Linear Independence, Basis and Dimension,
2.4 Four Fundamental Subspaces
2.5 Orthogonality - Orthogonal Vectors and Subspaces,
2.6 Cosines and Projections onto Lines
2.7 Orthogonal Bases and Gram – Schmidt

Unit 3	**Probability and Distributions**	**7 Hours**

3.1 Random variables. 3.2 Discrete distributions and Continuous distributions 3.3 Binomial Distribution 3.4 Poisson Distribution 3.5 Normal Distribution		
Unit 4	**Statistical Techniques**	**8 Hours**
4.1 Lines of regression of bivariate data, Correlation coefficient. 4.2 Fitting of Curves by method of Least-squares. 4.3 Fitting of Straight lines. 4.4 Fitting of Parabola. 4.5 Fitting of Exponential curves.		
Unit 5	**Introduction to Fuzzy sets**	**7 Hours**
5.1 Crisp Sets: An overview. 5.2 Fuzzy sets: Basic concepts 5.3 Operations on fuzzy sets. 5.4 Extension Principle		
Unit 6	**Fuzzy Arithmetic**	**6 Hours**
6.1 Fuzzy numbers. 6.2 Fuzzy cardinality 6.3 Operations on Fuzzy numbers. 6.4 Fuzzy equations of type A + X = B and A.X = B.		
Text Books:		
1. Higher Engineering Mathematics by Dr. B. S. Grewal. 2. Fuzzy sets and Fuzzy Logic by George J. Klir, Bo Yuan.		
Reference Books:		
1. Probability and Statistics for Computer science by James L. Johnon. 2. Fundamentals of Mathematical Statistics by Gupta and Kapoor. S. Lang, Introduction to Linear Algebra, 2nd Ed.,		

			L	T	P	Credit
Course Code:	UCSC0302		**3**	**1**		**4**
Course Name:	Discrete Mathematical Structures					

Course Prerequsites:	Mathematics - Probability theory, Set theory, functions

Course Description:	This Course consists of concepts of Discrete mathematical structures such as mathematical logic, Sets, relations, functions, lattices and Boolean algebra, combinatory and graph theory

Course Outcomes:

CO1	Explain the basic concepts of discrete mathematical structures
CO2	Demonstrate the applications of discrete structures in different fields of computer science.
CO3	Solve problems using the concepts of Discrete structures.
CO4	Apply the mathematical proofs and techniques to prove the theorems in computer science.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination (MSE)	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1	**Mathematical logic (Text book-1)**	**8 Hours**

1.1 Statements and Notations
1.2 Connectives , Statement formulas and truth tables, well formed formulas, Tautologies, Equivalence of formulas, Duality law, Tautological implications, functionally complete sets of connectives, other connectives
1.3 Normal and principal normal forms, completely parenthesized infix and polish notations
1.4 Theory of Inference for statement calculus – validity using truth table, rules of inference, consistency of Premises and indirect method of proof, Predicate calculus

Unit 2	**Set theory (Text book-1)**	**8 Hours**

2.1 Basic concepts of set theory, Operations on sets, Ordered pairs, Cartesian Products
2.2 Representation of discrete structures
2.3 Relation and ordering - properties of binary relations in a set, Relation matrix and the graph of a relation, Partition and Covering of set, Equivalence relations, Recurrence relations, Composition of Binary relations, Partial ordering , POSET and Hasse diagram.
2.4 Functions – types, composition of functions, Inverse functions.

Unit 3	**Algebraic systems (Text book-1)**	**5 Hours**

3.1 Algebraic systems, properties and examples
3.2 Semigroups and Monoids, properties and examples, Homomorphism of Semigroups and Monoids
3.3 Groups: Definition and examples, Subgroups and homomorphism

Unit 4	**Lattices and Boolean algebra (Text book-1)**	**5 Hours**

4.1 Lattice as POSETs , definition , examples and properties
4.2 Lattice as algebraic systems, Special lattices
4.3 Boolean algebra definition and examples
4.4 Boolean functions

Unit 5	**Permutations, Combinations and Probability theory (Text book-2)**	**7 Hours**

5.1 The Basics of Counting
5.2 The Pigeonhole Principle
5.3 Permutations and Combinations
5.4 Generalized Permutations and Combinations
5.5 Discrete Probability
5.6 Conditional probability
5.7 Bayes’ Theorem

Unit 6	**Graphs (Text book-2)**	**7 Hours**

6.1 Introduction to Graphs
6.2 Graph Terminology
6.3 Representing Graphs and Graph Isomorphism
6.4 Connectivity
6.5 Euler and Hamilton Paths
6.6 Planar Graphs
6.7 Introduction to Trees

Text Books:

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

Reference Books:

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

		L	T	P	Credit
Course Code:	UCSC0303				
Course Name:	Data Structures	3			3

Course Prerequsites:

Fundamentals of Programming Language.

Course Description:

Introduces data structure concepts like lists, stack, queues, trees, and graphs. Discussess about the implementations of these data objects, programming styles, and run-time representations. Examines algorithms for sorting, searching and some graph algorithms. Algorithm analysis and efficient code design is introduced.

Course Outcomes:

After completion of the course, students shall be able to -

CO1	explain various concepts of data structures.
CO2	analyze different data structures and algorithms to find their complexity.
CO3	select appropriate data structure(s) to solve different computing problems.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination (MSE)	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1		**6 Hours**

Concept of data, data structures, and data types; Abstract Data Types (ADT) - Atomic & Composite, Operations; Linear and non-linear data structures; Pseudo code; algorithm efficiency.

Unit 2		**8 Hours**

Concept of linked data organization; Types & representation of - singly linked list, doubly linked list, and circular linked list; Operations on lists - insertion, deletion, traversal, search, etc; Applications using these data structures.

Unit 3		**6 Hours**

Stack: Introduction; representation; opearations; implementation using array & list; applications of stack.
Queue: Introduction; representation; operations; implementation using array & list; types of queue - circular queue, double ended queue ,and priority queue; applications of queue.

Unit 4		**7 Hours**

Tree: Basic terminology; binary tree and its representation; binary tree traversal methods; binary search tree (BST), AVL tree, Heaps; Operations and applications of BST, AVL, Heaps

Unit 5 **5 Hours**

Graph: Basic terminology; Graph storage structures - adjacency matrix and adjacency list; Graph operations, graph traversal techniques - BFS, DFS; Applications using graphs

Unit 6 **8 Hours**

Searching and Sorting: 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.

Text Books:

1. Data Structures and Pseudocode approach with C, 2nd Edition by Richard F. Gilberg & Behrouz A. Forouzan
2. Data Structures using C and C++ by Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum
3. Data Structures by Seymour Lipschutz

Reference Books:

1. Data Structures and Algorithms by G A V Pai (McGrawHill)
2. Data Structures and Algorithms in python Michael T. Goodrich (Wiley)

Course Code:	UCSC0304	**L**	**T**	**P**	**Credi**
Course Name:	Digital Logic Design & Microprocessors	**3**			**3**

Course

Fundamentals of Electronics and Computers, Basic Number System and Boolean Algebra

Course Description:

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

Course Outcomes:

CO1	Describe working of basic digital components
CO2	Illustrate different microprocessors operations & addressing modes
CO3	Analyze changes in microprocessor evolution
CO4	Develop Assembly Language Programs

CO-PO Mapping:

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

Assessment Scheme:

SN	**Assessment**	**Weightage**	**Remark**
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination (MSE)	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1	**Combinational & Sequential Logic Design**	**7 Hours**

Number System, Boolean Algebra-Reduction, minterm,Maxterm,Multiplexer, implementation of expression using MUX, Demultiplexer, decoder(74138), Classification, Flip-flops(S-R, J-K,T,D)using gates, Race around condition Master –Slave J-K Flip Flop, Counters (Asynchronous & Synchronous), Design examples, Shift registers , State transition diagram, excitation table.

Unit 2	**8085 Microprocessor Architecture**	**7 Hours**

The 8085 MPU, Microprocessor communication and bus timing, Demultiplexing address and Data bus, Generating control signals, The 8085 Architecture, opcode fetch machine cycle, memory read and write machine cycle. 8085 instruction groups, addressing modes.

Unit 3	**8085 Programming Techniques**	**7 Hours**

Writing and execution assembly language program, counters & delays, Stack, Instruction related to stack execution of CALL and RET, The 8085 interrupt, RST instructions, vectored interrupts, RIM and SIM instructions
.
Basic interfacing concepts, peripherals I/O instructions IN, OUT, I/O execution, Memory - structure, interfacing & address decoding. Memory mapped I/O, I/O mapped I/O.

Unit 4	8086 Microprocessor and Assembly Language	9 Hours

Architecture of 8086, Registers of 8086, Memory Model, Addressing Modes, Instruction Set, Programming

Unit 5	80x86 Family	5 Hours

Introduction to 80186,80286,80386,80486 processors, Special Registers

The 80386 Microprocessor: The memory System, Special 80386 Registers
Virtual 8086 Mode, The Memory Paging Mechanism

Unit 6	Introduction to Pentium Microprocessors	5 Hours

The Pentium Microprocessor : The Memory System, Special Pentium Registers, Pentium Memory Management,
The Pentium Pro Microprocessor : Internal structure of the Pentium Pro
The Pentium 4 : Memory Interface, Register Set, Hyper Threading Technology.

Text Books:

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

Reference Books:

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

		L	T	P	Credit
Course Code:	UCSC0305				
Course Name:	Computer Network	**3**			**3**

Course Prerequsites:

Must have basic knowledge of computers and network

Course Description:

This course provides a solid understanding of OSI reference model and TCP/IP protocol suite. Also make us familiar with functionalities of different layers and gives exposure to different application layer protocol. This course will help students ready with all fundamental networking concepts

Course Outcomes:

CO1	Define different concepts of OSI/TCP/IP network models and physical layer
CO2	Make use of framing, error control, flow control and medium access control techniques
CO3	Elaborate IP addresses, IP protocols, types of routing algorithm and congestion control techniques
CO4	Describe process to process communication, multiplexing and transport layer protocols
CO5	Outline different types of application layer protocols from TCP/IP protocol suite

CO-PO Mapping:

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

Assessment Scheme:

SN	**Assessment**	**Weightage**	**Remark**
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination (MSE)	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1	**Introduction to Network**	**5 Hours**

1.1 Data Communication
1.2 Networks
1.3 Internet
1.4 Protocols and Standards
1.5 Layered Task
1.6 OSI Model and Layers
1.7 TCP/IP Protocol Suite
1.8 Addressing
1.9 Physical Layer and Media

Unit 2	**Data Link Control Layer**	**9 Hours**
2.1 Error Detection and Correction 2.2 Block Coding, Linear Block Codes 2.3 Cyclic Codes 2.4 Checksum 2.5 Data Link Control: Framing 2.6 Flow and Error Control 2.7 Protocols: Noiseless channels, Noisy Channels		
Unit 3	**Medium Access Control Sub layer**	**7 Hours**
3.1 Channel allocation Problem 3.2 Multiple Access Protocols: ALHOA, CSMA 3.3 Collision free protocols 3.4 Limited contention protocols 3.5 IEEE Standard 802 for LANS and MANS 3.6 Bridges 3.7 Introduction to VLANS		
Unit 4	**Network Layer**	**8 Hours**
4.1 Network Layer Design Issues Routing Algorithms : Shortest Path, Flooding, Distance Vector, Link State, Broadcast 4.2 IP,ARP,RARP,ICMP,IGMP 4.3 Congestion control algorithms: Principles, Congestion prevention policies, Traffic Shaping, congestion control in datagram subnet, Choke Packet, Load Shedding, Jitter Control 4.4 IPv4 Addresses : Introduction, Classfull and Classless addressing, Special Addresses and NAT		
Unit 5	**Transport Layer**	**4 Hours**
5.1 Transport Layer functions 5.2 UDP- datagram, services, applications 5.3 TCP - services, segment, connection, state transition diagram, Flow control, congestion control, error control, timers.		
Unit 6	**Application Layer**	**9 Hours**
6.1 DHCP: Introduction, Previous Protocols, DHCP operation, Packet Format, DHCP Configuration. 6.2 DNS: Need, Name Space, Domain Name Space, Distribution of name space, and DNS in internet, Resolution, DNS massages, Types of records, Compression examples, encapsulation. 6.3 Telnet and SSH 6.4 FTP and TFTP. 6.5 HTTP and SMTP 6.6 SNMP: Concept and Management Component,SMI,MIB,SNMP, UDP Port and Security		
Text Books:		
1. Data Communications and Networking – Behrouz A Forouzan (The McGraw Hill) (Unit 1,2,3) 2. Computer Networks – Andrew S. Tanenbaum- (Prentice Hall) 5th Edition (Unit 3, 4) 3. TCP/IP Protocol Suite- Behrouz Forouzan-(The McGraw Hill) (4,5,6)		

Reference Books:

1. Computer Networking with Internet Protocols and Technology, William Stallings (Prentice Hall)

Course Code:	UCSC0306	**L**	**T**	**P**	**Credit**
Course Name:	Data Structures Lab			2	**1**

Course Prerequsites:

Fundamentals of Programming Language.

Course Description:

The course have assignements based on the Data Structure and algorithms. The assignments will help students to understand the working of linear and non-linear data structures. Students will also focus on the applications of Data Structures and algorithms in computer programming.

Course Outcomes:

After completion of the course, students shall be able to -

CO1	Implement Linear and Non-Linear Data Structures
CO2	Implement Searching and Sorting algorithms
CO3	Develop programs and applications using data structures and algorithms.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	50%	Assignment, Test, Quiz, Seminar, Presentation,etc.
2	End Semester Examination (ESE)	50%	Practical Performance & Viva

Course Contents:

Assessment No. 1	**2 Hours**
Write a program using recursion techniq - 1. GCD calcualtion 2.Factorial Calculation 3. String Operations. Calculate the time complexity of the program.	
Assessment No. 2	**2 Hours**
Linear list using static memory allocation.	
Assessment No. 3	**2 Hours**
Create a Linear List by using Structure, memory management functions and pointers in C	
Assessment No. 4	**2 Hours**
Implement a Doubly Linked List and its all operations (Insert,Delete,Search)	
Assessment No. 5	**2 Hours**
Application of List - Write a program to maintain the navigational history (forward/backward) tracking	
Assessment No. 6	**2 Hours**
Implement the stack and its operation (array/list)	
Assessment No. 7	**2 Hours**
Implement a Circular Queue and its operation(array/list)	
Assessment No. 8	**2 Hours**
Write a program to convert infix expression into postfix expression	

Assessment No. 9	**2 Hours**
Write a program to implement Linear Search algorithm. Analyse the time complexity	
Assessment No. 10	**2 Hours**
Write a program to sort the array in ascending order. (use – Selection Sort/Bubble Sort/Insertion Sort)	
Assessment No. 11	**2 Hours**
Write a program to sort the array in ascending order.(Quick Sort) with recursion	
Assessment No. 12	**2 Hours**
Implement a binary search algorithm using array/list	
Assessment No. 13	**2 Hours**
Impement hashing algorithm using list and colision resolution.	

			L	T	P	Cred
Course Code:	UCSC0307					
Course Name:	Digital Logic Design and Microprocessor Lab				**2**	**1**

Course Prerequsites:

Digital Logic Design & Microprocessors

Course Description:

This subject covers practical details of subject Digital Logic Design and Microprocessors.

Course Outcomes:

CO1	Model basic digital circuits
CO2	Develop simple assembly language programs using 8085 instruction set

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1	PO1	PO1	PSO	PSO
CO1	2													
CO2		2												

Assessment Scheme:

SN	**Assessment**	**Weightag**	**Remark**
1	In Semester Evaluation (ISE)	100%	Experiment, Practical Performance and Oral
2	Practical Oral and Exam	100%	Oral Exam

Course Contents:

Experiment 1	**Study of MUX** **Aim and Objectives: Understand working of MUX** **Outcomes: Students will be able to implement MUX** **Theoretical Background: MUX -working, functions** **Experimentation: Construct MUX** **Results and Discussions: Truth Tables for MUX** **Conclusion: Implemented MUX**	**2 Hours**
Experiment 2	**Study of DEMUX** **Aim and Objectives: Understand working of DEMUX** **Outcomes: Students will be able to implement DEMUX** **Theoretical Background: DEMUX -working, functions** **Experimentation: Construct DEMUX** **Results and Discussions: Truth Tables for DEMUX** **Conclusion: Implemented DEMUX**	**2 Hours**

Experiment 3	**Study of R-S flip-flops** **Aim and Objectives: Construct R-S flip-flops** **Outcomes: Students will be able to implement R-S flip-flops** **Theoretical Background: Characteristics of different Flip-Flops** **Experimentation: Construct R-S flip-flops** **Results and Discussions: Truth Tables for R-S flip-flops** **Conclusion: Implemented circuit for R-S flip-flop**	**2 Hours**
Experiment 4	**Study of J-K flip-flops** **Aim and Objectives: Construct J-K flip-flops** **Outcomes: Students will be able to implement J-K flip-flops** **Theoretical Background: Characteristics of different Flip-Flops** **Experimentation: Construct J-K flip-flops** **Results and Discussions: Truth Tables for J-K flip-flops** **Conclusion: Implemented circuit for J-K flip-flop**	**2 Hours**
Experiment 5	**Study of Registers** **Aim and Objectives: Construct Different type of Registers** **Outcomes: Students will be able to implement Registers** **Theoretical Background: Characteristics of different Registers** **Experimentation: Construct Registers** **Results and Discussions: Truth Tables for Registers** **Conclusion: Implemented circuit for Registers**	**2 Hours**
Experiment 6	**Study of counters** **Aim and Objectives: Implementing UP and DOWN counter** **Outcomes: Students will be able to implement UP and DOWN counter** **Theoretical Background: Characteristics and types of counter** **Experimentation: Construct UP and DOWN counter** **Results and Discussions: Truth Tables for UP and DOWN counter** **Conclusion: Implemented circuit for UP and DOWN counter**	**2 Hours**

Experiment 7	Interfacing counter circuit with seven segment display Aim and Objectives: Interfacing counter circuit and seven segment display Outcomes: Students will be able to connect counter circuit to seven segment display Theoretical Background: Working of seven segment display Experimentation: Build interface for counter circuit and seven segment display Results and Discussions: Observation of output on seven segment display Conclusion: Built interface for counter circuit and seven segment display	2 Hours
Experiment 8	Study of 8085 microprocessor Aim and Objectives: Understand working of 8085 microprocessor Outcomes: Students will be able to explain instructions of 8085 microprocessor Theoretical Background: Architecture & Instruction Set of 8085 microprocessor Experimentation: Use various instructions of 8085 microprocessor in simulator Results and Discussions: Table of Instructions with purpose, mnemonic & size Conclusion: Demonstrated instructions using simulator	2 Hours
Experiment 9	Assembly language programming for 8085 Aim and Objectives: Writing simple assembly language programs (4 to 6) Outcomes: Students will be able to develop simple assembly language programs Theoretical Background: Instruction Set of 8085 microprocessor Experimentation: Develop algorithm and program for given problem statements Results and Discussions: Execute developed programs and note the results Conclusion:	2 Hours
Experiment 10	Study of 8086 microprocessor Aim and Objectives: Understand working of 8086 microprocessor Outcomes: Students will be able to explain instructions of 8086 microprocessor Theoretical Background: Architecture & Instruction Set of 8086 microprocessor Experimentation: Use various instructions of 8086 microprocessor in simulator Results and Discussions: Table of Instructions with purpose, mnemonic & size	2 Hours

Experiment 11	**Assembly language programming for 8086** **Aim and Objectives: Writing simple assembly language programs (4 to 6)** **Outcomes: Students will be able to develop simple assembly language programs** **Theoretical Background: Instruction Set of 8086 microprocessor** **Experimentation: Develop algorithm and program for given problem statements** **Results and Discussions: Execute developed programs and note the results** **Conclusion:**	**2 Hours**

Text Books:

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

Reference Books:

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

			L	T	P	Credit
Course Code:	UCSC0308					
Course Name:	Computer Network Lab				**2**	**1**

Course Prerequsites:

Must have basic knowledge of computers and Computer Network

Course Description:

This course provides a solid understanding of implementation of different framing, error control, flow control and routing algorithms. Help students to design network as per the requirement. Students can develop client server application using socket API and make them understand different application layer protocol with help of simulation and demonstration.

Course Outcomes:

CO1	Build sample network and VLAN as per the organization requirements
CO2	Develop software programs for framing, error control, flow control and routing algorithms
CO3	Make use of socket API to develop client-server programs
CO4	Inspect working of different types of application layer protocols from TCP/IP protocol suite
CO5	Develop FOSS server to configure different types of network services

CO-PO Mapping:

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

Assessment Scheme:

SN	**Assessment**	**Weightage**	**Remark**
1	In Semester Evaluation (ISE)	100%	Experiment, Practical Performance and Oral Exam
2	Practical Oral and Exam (POE)	100%	Practical Performance and Oral Exam

Course Contents:

Experiment 1	**Design and simulation of sample network**	**2 Hours**
Experiment 2	**Demonstration of network testing tools**	**2 Hours**
Experiment 3	**Implementation of framing techniques** **A) Character count** **B) Bit stuffing**	**2 Hours**
Experiment 4	**Implementation of Error control mechanisms** **A) CRC** **B) Hamming Code**	**2 Hours**
Experiment 5	**Implementation of Flow control mechanisms** **A) Stop and wait ARQ** **B) Go Back N** **C) Selective repeat**	**2 Hours**
Experiment 6	**Design and simulate working of Virtual LAN**	**2 Hours**

Experiment 7	**Implementation of Routing algorithm** **A) Shortest path routing** **B) Distance vector routing**	**2 Hours**
Experiment 8	**Implementation of Client-Server model** **A) Simple client-server model** **B) Iterative client-server model** **C) Concurrent client-server model**	**2 Hours**
Experiment 9	**Simulation of application layer protocol**	**2 Hours**
Experiment 10	**Installation and Configuration of FOSS server**	**2 Hours**

Text Books:

1. Data Communications and Networking – Behrouz A Forouzan (The McGraw Hill) (Unit 1,2,3)
2. Computer Networks – Andrew S. Tanenbaum- (Prentice Hall) 5th Edition (Unit 3, 4)
3. TCP/IP Protocol Suite- Behrouz Forouzan-(The McGraw Hill) (4,5,6)

Reference Books:

1. Computer Networking with Internet Protocols and Technology, William Stallings (Prentice Hall)

Course Code:	UCSC0309		**L**	**T**	**P**	**Credit**
Course Name:	Fundamentals of Web		**1**		**1**	**2**

Course Prerequsites

No prerequisites or basic understanding of programming.

Course Description:

This course is intended to teach students the fundamentals of web development in a Project Based Learning (PBL) environment. Students are taught and guided on the basic elements of web development: design, development, and hosting of the website.

Course Outcomes:

CO1	Design web pages using HTML and CSS
CO2	Develop responsive website using bootstrap
CO3	Developing interactive website using Jquery and Javascript
CO4	Design and Host the Website

CO-PO Mapping:

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

Assessment Scheme:

SN	**Assessment**	**Weightage**	**Remark**
1	In Semester Evaluation 1 (ISE1)	25M	Assignment, Test, Quiz, Seminar, Presentation, PBL, etc.
2	In Semester Evaluation 2 (ISE2)	25M	Assignment, Test, Quiz, Seminar, Presentation, PBL, etc.

Course Contents:

Unit 1	**HTML 5.0**	**3 Hours**

Introduction to HTML5, Features of HTML5, HTML5 DocType, New Structure Tags, Section, Nav, Article, Aside, Header, Footer, Designing a HTML Structure of Page, New Media Tags, Audio Tag, Video Tag, Canvas and Svg Tag, Introduction to HTML5 Forms, New Attributes, Placeholder Attribute, Require Attribute, Pattern Attribute, Autofocus Attribute, email , tel, url types, number type, date type, range type, voice search, Examples of Form

Unit 2	**CSS 3.0**	**3 Hours**

Introduction to CSS 3, New CSS 3 Selectors, Attribute Selectors, First-of-type, Last- of-type, Nth-child, Element:empty, New CSS3 Properties, Custom Fonts, Text- Shadow Property, Text-Stroke Property, Rounded Corners, Box Shadows, CSS Gradients, CSS Multiple backgrounds, Opacity Property, Transition effect, Transform effect, Animation effects, Css Media Queries, Using CSS3 in Practical Layout

Unit 3	**BootStrap**	**2 Hours**
Introduction to Responsive Design, Mobile first design concepts, Common device dimensions, View-port tag, Using css media queries, Menu conversion script, Basic Custom Layout, Introduction to Bootstrap, Installation of Bootstrap, Grid System, Forms, Buttons, Icons Integration, Using CSS3 in Practical Layout		
Unit 4	**JavaScript**	**2 Hours**
Introduction to Client Side Scripting, Introduction to Java Script, Javascript Types, Variables in JS, Operators in JS, Conditions Statements, Java Script Loops, JS Popup Boxes, JS Events, JS Arrays, Working with Arrays, JS Objects, JS Functions, Using Java Script in Realtime, Validation of Forms, Related Examples, Frameworks of js.		
Unit 5	**jQuery and jQuery UI**	**2 Hours**
Introduction to jQuery, jQuery Features, Installing jQuery, jQuery Syntax, jQuery Ready Function, jQuery Selectors, jQuery Actions, jQuery plugins, jQuery Validation plugin, jQuery Slideshow, jQuery Dropdown, jQuery UI, Working with jQueryUI, jQuery Accordions, jQuery Tabs, jQuery Tooltips, jQuery Autocomplete		
Unit 6	**Web Hosting**	**2 Hours**
Web Hosting Basics, Types of Hosting Packages, Registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel, Using FTP Client, Maintaining a Website, Introduction to Joomla & Wordpress CMS		
Text Books:		
1. HTML & CSS: The Complete Reference, Fifth Edition by ThomasPowell 2. JavaScript: The Definitive Guide, 6th Edition ByDavid Flanagan 3. Learning jQuery Fourth Edition byJonathan Chaffer,KarlSwedberg		
Reference Books and Resources:		
1. HTML & CSS: The Complete Reference, Fifth Edition, by Thomas Powell 2. JavaScript: The Definitive Guide, 6th Edition, by David Flanagan 3. Learning jQuery Fourth Edition, by Jonathan Chaffer & Karl Swedberg 4. https://www.w3schools.com/		

SEM-II

			L	T	P	Credit
Course Code:	UCSC0401		**3**	**1**		**4**
Course Name:	Computer Algorithm					

Course Prerequsites:	Data Structures

Data Structures

Course Description:

This course introduces fundamental concepts and key techniques for designing and analyzing algorithms along with studying and applying different algorithm design methods namely, greedy method, divide and conquer, dynamic programming and backtracking.

Course Outcomes:

CO1	Define basic concepts of algorithms and measure the efficiency of algorithm.
CO2	Make use of standard design techniques such as divide and conquer, greedy algorithms, dynamic
CO3	Identify graph algorithms to model real life engineering problems.
CO4	Distinguish between P and NP Classes of problems.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightag	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1	**Introduction:**	**6 Hours**

What is algorithm, Algorithm Specification: Pseudocode Conventions, Recursive Algorithm, Performance Analysis: Space Complexity, Time Complexity, Asymptotic Notations, Practical Complexities, Performance Measurement Recurrences: The substitution method, recursion tree method

Unit 2	**Algorithm Design and Analysis Techniques – I:**	**8 Hours**

Divide and Conquer-The general method, Binary search, Finding the maximum and minimum, Merge sort, Quick sort and analysis of these algorithms.
The Greedy method: The general method, Knapsack problem, Job sequencing with deadlines, Optimal storage on tapes, Optimal merge patterns, Huffman codes.

<table>
<tr><td><b>Unit 3</b></td><td><b>Algorithm Design and Analysis Techniques - II:</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3"><b>Dynamic Programming:</b> The general method, Multistage graphs, Optimal binary search trees, 0/1 knapsack, Reliability design, Traveling Salesperson problem.</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td><b>Unit 4</b></td><td><b>Graph Algorithms: Elementary Graph Algorithms:</b></td><td><b>9 Hours</b></td></tr>
<tr><td colspan="3">Representations of graphs , Breadth-first search, Depth first search, Strongly connected components, Minimum Spanning Trees: Growing a minimum spanning tree, The algorithms of Kruskal and Prim
Single-Source Shortest Paths: The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra’s algorithm, The Floyd-Warshall algorithm</td></tr>
<tr><td><b>Unit 5</b></td><td><b>Backtracking:</b></td><td><b>7 Hours</b></td></tr>
<tr><td colspan="3">The general method, 8-queen problem, Sum of subsets, Graph Coloring, Knapsack Problem, Hamiltonian Cycle.</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td><b>Unit 6</b></td><td><b>Complexity classes :P & NP-Complete:</b></td><td><b>5 Hours</b></td></tr>
<tr><td colspan="3">Polynomial time, Polynomial-time verification, Decidability , NP completeness and reducibility, NP-complete problems ,string matching algorithms, case studies</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"><b>Text Books:</b></td></tr>
<tr><td colspan="3">1. Thomas Cormen, Charles Leiserson, Ronald Rivest and Cliford Stein, “Introduction to Algorithms”, PHI
2. Fundamentals of Computer Algorithms - Ellis Horowitz, Satraj Sahani, Saguthevar Rajasejaran, Universities Press, Second Edition.</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"><b>Reference Books:</b></td></tr>
<tr><td colspan="3">1. Fundamentals of Algorithmics – Gilles Brassard, Paul Bratley (Pearson Education).
2. Mastering Algorithms with C – Kyle Loudon (SPD O’Reilly).
3. Computer Algorithms- Introduction to Design and Analysis – Sara Baase, Allen Van Gelder (Pearson Education).</td></tr>
</table>

			L	T	P	Credi
Course Code:	UCSC0402		**3**	**1**		**4**
Course Name:	Automata Theory					

Course

Discrete Mathematics, Sets, Cartesian Product and Functions

Course

This course deals with the theoretical background of computer science.

Course Outcomes:

CO1	Explain types of formal languages and their acceptors
CO2	Classify formal languages on the basis of their features
CO3	Relate the computational models with the modern day computer technologies
CO4	Design computational machines of various types for specified problems

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination	50%	100% course contents

Course Contents:

Unit	**Mathematical Induction, Regular Languages & Finite Automata**	**8 Hours**

The Principle of Mathematical Induction Recursive Definitions, Definition & types of grammars & languages, Regular expressions and corresponding regular languages,
examples and applications, unions, intersection & complements of regular languages, Finite automata-definition and representation, on-deterministic F.A.,NFA with null
transitions, Equivalence of FA's , NFA's and NFA's with null transitions.

Unit	**Kleene's Theorem**	**4 Hours**

Part I & II statements and proofs, minimum state of FA for a regular language,
minimizing number of states in Finite Automata.

Unit	**Grammars and Languages**	**10 Hours**

Derivation and ambiguity, BNF & CNF notations, Union, Concatenation and *'s of
CFLs, Eliminating production & unit productions from CFG, Eliminating useless
variables from a context Free Grammar. Parsing: Top-Down, Recursive Descent and
Bottom-Up Parsing

Unit	**Push Down Automata**	**4 Hours**
Definition, Deterministic PDA & types of acceptance, Equivalence of CFG's & PDA's.		
Unit	**CFL's and non CFL's**	**4 Hours**
Pumping Lemma and examples, intersections and complements		
Unit	**Turing Machines**	**10 Hours**
Models of computation, definition of Turing Machine as Language acceptors, combining Turing Machines, Computing a function with a TM, Non-deterministic TM and Universal TM, Recursively enumerable languages, Unsolvable problems.		
Text Books:		
1.Introduction to languages & Theory of computations – John C. Martin (MGH) –Chapters 1, 2,3,4,5,6,7,8 2. Discrete Mathematical Structures with applications to Computer Science—J .P.Trembley & R.Manohar (MGH) Chapter 1,		
Reference Books:		
1.Introduction to Automata Theory , Languages and computation – John E. Hopcraft , Rajeev Motwani , Jeffrey D. Ullman (Pearson Edition). 2.Introduction to Theory of Computations – Michael Sipser (Thomson Brooks / Cole) 3.Theory Of Computation- Vivek Kulkarni, 1st edition OXFORD university Press 4.Theory Of Computation A problem Solving Approach Kavi Mahesh Wiley India		

			L	T	P	Credi
Course Code:	UCSC0403					
Course Name:	Computer Graphics		**3**			**3**

Course

Course

Study basic and core concepts in Computer Graphics

Course Outcomes:

CO1	Explain the basic concepts of interactive computer graphics.
CO2	Illustrate the fundamental concepts of computer graphics using mathematical models and
CO3	Analyze basic illumination models and polygon rendering methods.
CO4	Build Animation sequences.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit	Introduction	5 Hours

1.1 Overview of graphics systems – Video display devices,
1.2 Raster scan systems
1.3 Random scan systems
1.4 Input and Output Device

Unit	Transformations	9 Hours

2.1 Basic 2D & 3D transformations - Translation, Scaling, Rotation, Reflection, Shearing, Multiple Transformations
2.2 Translation and Homogeneous Coordinates
2.3 2D –Rotation about arbitrary point, reflection through an arbitrary line
2.4 3D - Rotation about an axis parallel to a coordinate axis, Rotation about an arbitrary axis in space, Reflection through an arbitrary plane
2.5 Windowing and View-porting,
2.6 Sutherland - Cohen line clipping algorithm

Unit	**Raster Scan Graphics**	**7 Hours**

3.1 DDA
3.2Bresenhams Line drawing algorithm
3.2 Bresenhams Circle drawing algorithm
3.3 Scan Conversion techniques: RLE, Frame Buffer
3.4 Scan converting polygons: Edge fill and Seed fill algorithms
3.5 Anti-aliasing

Unit	**Curves and Surfaces**	**8 Hours**

4.1 Non-parametric and parametric curves
4.2 Representation of space curves
4.3 Cubic Spline
4.4 Bezier curves
4.5 Z- buffer algorithm
4.6 Warnock algorithm

Unit	**Illumination models and surface rendering methods**	**4 Hours**

5.1 Light sources
5.2 Basic illumination models
5.3 Displaying light intensities
5.4 Halftone patterns and Dithering Techniques
5.5 Polygon Rendering methods
5.6 Ray tracing methods

Unit	**Computer Animation**	**9 Hours**

6.1 Introduction,
6.2 Key frame animation,
6.3 Construction of an animation sequence,
6.4 Motion control methods,
6.5 Procedural animation,
6.6 Key-frame animation vs. Procedural animation,
6.7 Introduction to Morphing, Wraping techniques,
6.8 Three dimensional morphing.

Text Books:

1. Computer Graphics C Version second edition –Donald D. Hearn, M. Pauline Baker (Pearson)
2. Mathematical elements for Computer Graphics - David F. Rogers, J. Alan Adams (MGH International)
3. Procedural elements for Computer Graphics - David F. Rogers (MGH International)
4. Computer Graphics- Rajesh Maurya (WILEY India)

Reference Books:

1. Principles of Computer Graphics Theory and Practice Using OpenGL and Maya, Shalini Govil-Pai, (Springer) .

2. Computer Graphics (second Edition) - Zhigang Xiang & Roy Plastock (Schaum's Outline Series, TMGH).

3. Computer Graphics Using OpenGL F.S. Hill Jr. Stephen M. Kelley, (Pearson Education).

			L	T	P	Credit
Course Code:	UCSC0404		**L**	**T**	**P**	**Credit**
Course Name:	Computer Organization and Architecture		**3**			**3**

Course Prerequsites:

Course Description:

Study basic and core concepts in Computer organization and advanced architectures

Course Outcomes:

CO1	Explain the organization of basic computer and its function, instruction types and data formats
CO2	Demonstrate the design of arithmetic unit and control unit
CO3	Evaluate cost, performance measures of computer system and design trade-offs
CO4	Discuss memory organization and memory management system
CO5	Explain the concepts of parallel, pipelined and distributed computer architectures

CO-PO Mapping:

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

Assessment Scheme:

SN	**Assessment**	**Weightage**	**Remark**
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation,
2	Mid Semester Examination	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation,
4	End Semester Examination	50%	100% course contents

Course Contents:

Unit 1 | Basic Computer Organization | 7 Hours

Evolution of computers - Electronic computers-generations, VLSI era , CPU organization , user and supervisor modes, accumulator based CPU, System bus, types of instruction(zero, one, two and three address machines), RISC& CISC, definition, comparison and examples. Data representation: Fixed-Point Numbers, Floating Point Number- The IEEE 754 floating pointing numbers.

Unit 2 | Arithmetic Design | 8 Hours

Fixed point arithmetic - high speed adders, adder expansion, Fixed point multiplication - Two’s complement multiplier, Booth’s algorithm, Combinational array multiplier, Fixed point division - Restoring, Non restoring algorithm, Combinational array divider, Division by repeated multiplication. Floating point arithmetic - Basic operations, Difficulties, Floating point units, Addition, subtraction, multiplication, division

Unit 3 | Control Unit Design | 7 Hours

Introduction, multi cycle operation, implementation methods, Hardwired control, design methods, state tables, GCD processor, Classical method, one hot method. Microprogrammed control unit organization, parallelism in microinstructions, Microinstruction addressing, timing. Design example: twos complement multiplier control unit, Control field encoding, encoding by function, multiple microinstruction formats.

Unit 4 **Memory Organization** **9 Hours**

Types of memory, Memory systems, multi level, address translation, memory allocation, Caches, Associative memory, direct mapping, set associative addressing

Unit 5 **Introduction to Pipeline and Parallel Processing** **5 Hours**

Pipelining, linear pipelining, classification of pipeline processors Interleaved memory organization, performance evaluation factors. Parallel Processors Flynn’s Classification. Introduction to Associative memory processors

Unit 6 **Distributed Memory Architecture** **7 Hours**

Loosely coupled and tightly coupled architectures. Cluster computing as an application of loosely coupled architecture. Examples – CM*

Text Books:

1. Computer Architecture and Organization - John P Hayes (MGH) 3rd Edition
2. Advanced computer architecture – Kai Hwang(MGH)

Reference Books:

1. Computer Architecture & Parallel Processing – Kai Hwang & Briggs (MGH)
2. Computer Organization - Hamacher Zaky (MGH).

		L	T	P	Credit
Course Code:	UCSC0405				
Course Name:	Software Engineering	**3**			**3**

Course Prerequsites:

Course Description:

This course provides basic concepts, principles of software engineering & basics of project management

Course Outcomes:

CO1	Explain the software development process
CO2	Illustrate the software testing techniques and quality assurance in detail
CO3	Make use of project management concepts in project development
CO4	Design the solutions to the problems using Object Oriented Modelling with UML

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation 1 (ISE1)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
2	Mid Semester Examination (MSE)	30%	50% of course contents
3	In Semester Evaluation 2 (ISE2)	10%	Assignment, Test, Quiz, Seminar, Presentation, etc.
4	End Semester Examination (ESE)	50%	100% course contents

Course Contents:

Unit 1	**The Software Problem**	**5 Hours**

1.1 Software Problems, Software Engineering Problems 1.2 Cost, Schedule,Quality, Scale and Change 1.3 Software Development Process Modules 1.4 Project Management Process 1.5 Software Processes: Process & Project

Unit 2	**Requirement Analysis and Specifications**	**6 Hours**

2.1 Requirement gathering and analysis 2.2 Software Requirement Specifications 2.3 Collecting requirements and defining scopes 2.4 Creating the work break down structure, validating scope, controlling scope 2.5 Basic principles of cost management 2.6 Planning cost management , estimating costs 2.7Determing the budget, controling costs, formal system development techniques

Unit 3	**Design**	**7 Hours**

3.1 Design Concepts 3.2 Function Oriented Design 3.3 Object Oriented Design 3.4Detail Design 3.5 Verification 3.6 Metrics

Unit 4	Object Oriented Modelling and Design	6 Hours

4.1 Object Oeiented Design: What is object orientation? What is OO development? OO Themes 4.2 Modelling as Design Techniques: Modelling, Abstraction, Three Models 4.3 Overview of UML 4.5 Architecture

Unit 5	Agile Methodology	7 Hours

5.1 Introduction to Agile Methodology
5.2 Agile Software Development lifecycle
5.3 Agile Methodology - Scrum Methodology, Kanban Methodology
5.4 Agile Practices - Sustainable Pace, Story Mapping, Test Driven Development, Pair Programming, Unit Testing, Acceptance Testing, Agile Planning
5.5 Agile Metrics- BurnDown Chart, Lead Time & Cycle Time, Agile Velocity
5.6 Scaled Agile - Scaled Agile Frameworks

Unit 6	Quality Management	7 Hours

6.1 Importance, Planning Quality Management,
6.2 Performing Quality Assurance, Controlling Quality,
6.3 Tools and Techniques for Quality Control,
6.4 Modern Quality Management, Improving IT Project Quality
6.5 ISO 9000 SEI capability Maturity Model, Six Sigma
6.7 Agile Quality Management

Text Books:

1. Software Engineering: A precise Approach - Pankaj Jalote (Wiley India)
2. Information Technology Project Management, 7E, Kathy Schwalbe, Cengage Learning (India Edition)
3. Object Oriented Modeling and Design with UML, Michel R Blaha, James R Rambaugh, Second Edition
4. The Unified Modelling Language User Guide: - Grady Booch, James Rambaugh, Lvar Jacobson
5. Essential Scrum: A Practical Guide to the Most Popular Agile Process by S. Kenneth Rubin(India Edition)

Reference Books:

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

			L	T	P	Credit
Course Code:	UCSC0406					
Course Name:	Computer Graphics Lab				**2**	**1**

Course Prerequsites:

C programming and mathematics

Course Description:

Study and implement basic and core techniques in Computer Graphics.

Course Outcomes:

CO1	Develop programs to implement various transformations and algorithms in computer graphics.
CO2	Make use of modern tools such as blender, adobe flash, scratch, etc. for developing computer graphics
CO3	Show effects of various illumination models and ray tracing methods in computer graphics.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remark
1	In Semester Evaluation (ISE)	100%	Experiment, Practical Performance and Oral Exam
2	Practical Oral and Exam (POE)	100%	Practical Performance and Oral Exam

Course Contents:

Experiment 1	**Write a menu driven program to implement 2D/3D transformations.**	**2 Hours**

Experiment 2	**Implementing Bresenham's line drawing algorithm.**	**2 Hours**

Experiment 3	**Implementing Bresenham's circle generation algorithm.**	**2 Hours**

Experiment 4	**Implementing Edge fill algorithm.**	**2 Hours**

Experiment 5	**Implementing Seed fill algorithm.**	**2 Hours**

Experiment 6	**Implementing Sutherland-Cohen line clipping algorithm.**	**2 Hours**

Experiment 7	**Implementation of Bezier Curve.**	**2 Hours**
Experiment 8	**Implementing Basic Ray Tracing algorithm.**	**2 Hours**
Experiment 9	**3D object design using rendering tool like blender.**	**2 Hours**
Experiment 10	**Create animation using tools like blender, scratch.**	**2 Hours**
Experiment 11	**OpenGL programming to use basic graphics primitives**	**2 Hours**

Text Books:

Computer Graphics Using OpenGL F.S. Hill Jr. Stephen M. Kelley, (Pearson Education).

Reference Books:

			L	T	P	Credit
Course Code:	UCSC0407		**L**	**T**	**P**	**Credit**
Course Name:	Object Oriented Programming		**2**		**2**	**3**

Course Prerequsites:

C programming knwoledge

Course Description:

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

Course Outcomes: Student will be able to

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

CO-PO Mapping:

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

Assessment Scheme:

SN	**Assessment**	**Weightage**	**Remarks**
1	ISE	50%	In Semester Evaluation based on assigments, Quiz, Oral
4	ESE	50%	POE of 50 marks

Course Contents:

Unit 1	**Introduction and Basics of OOP**	**4 Hours**

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**	**4 Hours**

Variable declarations, global scope, const variables, reference variables, function prototypes, 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**	**4 Hours**
Need of Inheritance, Concept, public, private, protected inheritance, Single inheritance, Multiple and multilevel inheritance, Hybrid Inheritance, Virtual base class, overriding of member functions		
Unit 4	**Polymorphism**	**5 Hours**
Pointers basics of memory management, New and delete operators, Pointer to object, Pointer to data members, this pointer. Need of Polymorphism, concept, Compile time polymorphism or early binding: function overloading and operator overloading, operator overloading using member function and friend function, overloading - unary, binary, arithmetic operators, relational operators, Overloading new and delete operators, insertion and extraction operators, Run time polymorphism or late binding using Virtual function, pure virtual function, Abstract class, Type conversion		
Unit 5	**Files and Streams**	**4 Hours**
Concept of Streams, concept of File, opening and closing a file, detecting end-of-file, file modes, file pointer, reading and writing characters, strings and objects to the file, operations to move file pointers i.e. seekg, seekp, tellg, tellp.		
Unit 6	**Advanced C++ features**	**5 Hours**
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 Exception handling: Introduction, syntax for exception handling code: try-catch-throw, Multiple Exceptions, Exceptions with arguments		
Experiments based on		
1	Study of OOP features and compare it with POP	2 Hours
2	Functions with default (Optional) arguments.	2 Hours
3	Classes (with constructor) and Objects.	2 Hours
4	Operator Overloading.	2 Hours
5	Inheritance	2 Hours
6	Memory Management	2 Hours
7	Polymorphism	2 Hours
8	Type Conversion	2 Hours
9	Exception Handling	2 Hours
10	Template	2 Hours

11	File Handling	2 Hours
12	STL	2 Hours

Text Books:

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

Reference Books:

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

		L	T	P	Credit
Course Code:	**UCSC0408**				
Course Name:	**Mini Project-I**			**2**	**1**

Course Prerequisites:

Knowledge of Project Based Learning (PBL) concepts.

Course Description:

In this mini project, the students will apply Project Based Learning to a multi-course environment for solving different real-world problems. The students shall use the concepts they have learned in their S.Y. B.Tech Program (SEM-III) & the courses they are learning in the current semester i.e. SEM-IV. Students will develop a solution to an identified problem.

Course Outcomes:	Student should be able to
CO1	Identify real world problems which can be solved using CS concepts and technologies.
CO2	Describe the the proposed solution to the real world problem using technical report.
CO3	Implement the proposed solution using Computer Science & Engineering techniques.
CO4	Build detailed project report.

CO-PO Mapping:

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

Assessment Scheme:

SN	Assessment	Weightage	Remarks
1	ISE 1	50%	In Semester Evaluation based on Progress of the project
4	ISE 2	50%	In Semester Evaluation based on Progress of the project

Course Contents:

Guidelines for Mini Project -I

1 The primary objective of the mini project-I is to achieve multi course project based learning.

2 Course Instructor shall form the project team of 3 to 4 students in the batch of students

3 Each team shall use the knowledge they learned in the SY B.Tech courses to identify the real world problem which can be solved using technology

4 The solution shall be using the tools & techniques from multiple courses - e.g a solution shall be using data structures, networking algorithm, Web Technology to develop mini project

5 As students have undertaken Fundamentals of Web - its recommended to develop user interface using HTML

6 The evaluation shall be done in two phases

Phase 1	ISE-1	In ISE 1 the students shall be graded based on the skills demonstrated to identify the problem statement, define the problem statement & Designing its solution. The partial working model is expected to be completed.
Phase 2	ISE-2	In ISE 2 the students shall be graded based on the complete project implementation and its working. Followed by the detailed project report which shall cover the technical aspects of the project.

7 Its recommended to share a common project report format to all batches.

8 All course instructors shall coordinate and work towards common evaluation process.

9 Course instructors shall demonstrate and discuss sample case studies with students to help them understand the mini project deliverables.