Walchand College of Engineering, Sangli

(Government Aided Autonomous Institute)

Course Contents (Syllabus) for

Second Year B. Tech.
(Computer Science and Engineering)
Sem - III to IV

AY 2020-21

ODD Semester

Professional Core (Theory) Courses

Title of the Course: Probability Theory and Statistics **Course Code: 5MA201**	L	T	P	Cr
	2	0	0	2

Desirable requirements:

Textbooks:

Textbooks:

1.

References:

1.

Course Objectives:

1. .

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1			
CO2			
CO3			

CO-PO Mapping :

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

1: Low, 2: Medium, 3: High

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.

ISE 1 and ISE 2 are based on assignment/declared test/quiz/seminar/oral etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module wise Measurable Students Learning Outcomes :

-

Title of the Course: Discrete Mathematics **Course Code**: **5CS202**	L	T	P	Cr
	3	0	0	3

Desirable requirements: Mathematics-(set theory, Boolean operations, logical operations)

Textbooks:

1. J.P. Tremblay &R. Manohar , *"*Discrete Mathematical structure with applications to computer*",* McGraw Hill,1st Edition, 2001
2. Liu, *"*Elements of Discrete Mathematics*",* Tata McGraw Hill,3rd edition 2008
3. Kenneth Rosen, "Discrete Mathematics & its application" McGraw Hill, 7th edition 2012.

References:

1. K.D. Joshi, *"*Foundation of Discrete Mathematics*",* New Age International Ltd,1st edition,2014
2. Seymour Lipschutz , Marc Lipson *"*Discrete Mathematics: Schaum's Outlines Series*",* Schaum's outline series.,3rd edition, 2009

Course Objectives : This subject enhances one's ability to reason and ability to present a coherent and mathematically accurate argument. About 30% of the course time will be spent on logic, set theory, counting techniques and remaining 60% of the course time will be devoted to functions, relations, algebraic structures, graph theory, permutation and combination. Objectives of this course are as follows:

1. Deliver basic concepts of Logic theory to solve real life problems.
2. Introduce graphs, trees and algebraic structure and develop an attitude to solve problems based on these topics.
4. To give deep insight into discrete probability and combinatorics.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** logical notation to define and reason about fundamental mathematical concepts of logic theory, set theory, relations, probability, counting techniques.	2	Understanding
CO2	**Demonstrate** knowledge and skills obtained to investigate and solve problems of POSET, Hasse diagram, groups, semi group and monoid.	3	Applying
CO3	**Analyse** concepts and algorithms of graph theory and elementary combinatorial processes such as permutations and combinations.	4	Analysing

CO-PO Mapping :

PO and PSO	**PO1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**
CO2	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**
CO3	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**

1: Low, 2: Medium, 3: High

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 quiz/assignments.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1 Mathematical Logic & Set Theory	**6 Hrs.**
Introduction, Statement and Notation, Connectives, statements formulas and truth tables, well-formed formulas, Tautologies Equivalence of formulas, Tautologies, other connectives, Normal & Principal Normal forms. Basic concepts of set theory, Venn Diagram, set operation, algebra of sets.	
Module 2 Relations and Functions	**7 Hrs.**
Relations, Pictorial representation of Relations, Properties of binary relation, Equivalence Relations, partition and covering of set, POSET and Hasse Diagram, Functions - types, Inverse and composition of functions, lattice.	
Module 3 Algebraic structures	**6 Hrs.**
Introduction, Operations, semigroups, Groups, subgroups, Rings, monoid.	
Module 4 Graph theory and its applications	**7 Hrs.**
Basic terminology, multigraphs and weighted graphs, Paths and Shortest path in weighted graphs, Hamiltonian and Eulerian Paths and Circuits, Factor of a graph, Planner Graph.	
Module 5 Directed graphs	**6 Hrs.**
Trees, Rooted Trees, Path lengths in rooted trees, Prefix codes, Binary search trees, Spanning trees and cut sets, Minimal spanning trees, Kruskal's algorithm and Prim's algorithms, Warshall's algorithm for transitive closure.	
Module 6 Permutation, Combination and Discrete Probabilities	**7 Hrs.**
Basic counting techniques – inclusion and exclusion, Rules of sum and product, permutations, combinations, generation of permutations and combinations, Introduction to Discrete Probability, entropy and mutual information, recursion.	

Module wise Measurable Students Learning Outcomes :
After the completion of the course the student should be able to:

Module 1:
- Able to construct and explain logical proofs as logic plays a major role in formal languages and in hardware and software.

Module 2:
- Grasp concepts of relations and functions and demonstrate skills to solve related problems.

Module 3:
- Identify different algebraic structures

Module 4:
- Apply graph application in computer domain- e.g. finding shortest path in networking etc.

Module 5:
- Analyse concepts of trees, minimum cost spanning trees using different types of algorithms.

Module 6:
- Solve problems on permutation, combination and probability.

Tutorials: Group work solutions for Problems based on:
1. CNF, DNF, set theory, Venn Diagrams.
2. Equivalence relation, binary relation.
3. POSET and HASSE diagrams.
4. Algebraic structures.

5. Graph, Shortest path in weighted graphs ,Hamiltonian and Eulerian Paths and Circuits
6. Real Time Application of graph theory.
7. Warshall's Algorithm, prims and Kruskal's algorithm
8. Finding out optimal solution based on graph theory and trees.
9. Permutation and combination.

Title of the Course: Data Structures **Course Code: 5CS203**	L	T	P	Cr
	3	0	0	3

Desirable requirements: Programming in C including pointers and File Handling

Textbooks:

1. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures, A Pseudocode Approach With C",Cengage Learning, Second Edition, 2014
2. S. Lipschutz, "Data Structures,Schaum's" Outlines Series, Tata McGraw-Hill, 2013
3. Ellis Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++*",* Galgotia Book Source, New Delhi, 2008

References:

1. Yashavant Kanetkar, "Understanding pointers in C*",* BPB Publication, 4th Edition, 2009
2. N. B. Venkateshwarlu, E. V. Prasad, "C and Data Structures*",* S. Chand and Company, 2010
3. Jean-Paul Tremblay, Paul. G. Soresan, "An introduction to data structures with Applications*",* Tata Mc-Graw Hill International Editions, 2nd edition, 1984

Course Objectives :

This course is one of the core subject for Computer Science and Engineering students. The course mainly focuses on introducing various linear and nonlinear data structures, their characteristics and applications. The insight knowledge of various searching and sorting techniques enables the students to identify and apply suitable technique for different applications.

1. To make the students understand elementary linear and non-linear data structures and concepts of ADTs.
2. To develop and improve logical thinking and to make the students capable of applying appropriate data structure for modelling a given problem.
3. To provide a foundation to analyse and compare various searching and sorting techniques and to select appropriate technique to solve the problem.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** the fundamental concepts of structuring, managing and organizing the data using linear and non-linear data structures with ADTs, write recursive algorithms and explain various searching and sorting techniques	2	Understanding
CO2	**Choose** suitable data structure to be used and apply it to solve the various problems	3	Applying
CO3	**Compare** and **Analyze** various algorithms, searching and sorting methods based on inherent properties of data structures and the complexity of algorithms.	4	Analyzing

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**2**	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	**3**	**3**	**2**	-	-	-	-	-	-	-	-	-	**3**	-
CO3	**3**	**3**	**2**	-	-	-	-	-	-	-	-	-	**3**	-

1: Low, 2: Medium, 3: High

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 declared test/quiz/seminar
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Introduction	**6 Hrs.**
Basic Concepts: Algorithm, Pseudocode, ADT, Data Structure, Algorithmic Efficiency **Recursion:** Direct and Indirect recursion, analysis of recursive functions e.g. Towers of Hanoi, Ackerman's function, etc.	
Module 2: Linked Lists	**6 Hrs.**
Concept of linked organization, Singly linked list, doubly linked list and dynamic storage management, circular linked list, Operations such as insertion, deletion, inversion, concatenation, computation of length, traversal on linked list, Representation and manipulations of polynomials using linked lists.	
Module 3: Stacks and Queues	**6 Hrs.**
Fundamentals stack and queue as ADT, Representation and Implementation of stack and queue using sequential and linked organization, Circular queue: representation and implementation, Application of stack for expression evaluation and for expression conversion, Backtracking, Stacks and Recursion, Priority queue Doubly Ended Queue.	
Module 4: Trees	**7 Hrs.**
Basic terminology, binary trees and its representation, binary tree traversals (recursive and non-recursive), operations such as copy, equal on binary tree, expression trees, AVL Tree, Binary Search Trees, Heaps and its operations, Introduction to Multiway Trees.	
Module 5: Graphs	**5 Hrs.**
Terminology and Representation of graphs using adjacency matrix, adjacency list and adjacency Multilist, Traversals Depth First and Breadth First, Minimum Spanning Tree.	
Module 6: Searching & Sorting Technique	**9 Hrs.**
Search: Importance of searching, Sequential, Binary, Fibonacci search algorithms **Sorting:** Internal and External Sorts**,** Insertion, Shell, Heap, Quick sort, Merge sort, Radix sort, Two-way merge sort **Hashing:** Hashing functions, overflow handling with and without chaining, open addressing: linear, quadratic, double, rehashing **Introduction to Files and Indexes** (concept only implementation not expected): Indexing Techniques: hashed indexes, Tree indexing - B-trees File Organizations: Sequential, Random and Linked organizations.	

Module wise Measurable Students Learning Outcomes :

After the completion of the course the student should be able to:

Module 1:

- Explain ADT, build logic to solve the problem, write algorithms and think recursively.

Module 2:

- Apply concept of linked list and use of ADTs to solve the problem

Module 3:

- Choose data structures such as stacks and queues as the programmers' tool to solve problems.

Module 4:

- Apply non-linear data structure, tree and its basic operations and use it to solve the problem.

Module 5:

- Explain and use graph as a data structure to store and manipulate data for various applications

Module 6:

- Explain hashing, file organizations and compare various searching and sorting techniques.

Title of the Course: Data Communication Course Code: **5CS204**	L	T	P	Cr
	3	0	0	3

Desirable Requirements : Nil

Textbooks:

1. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw-Hill, 4th/5th Edition, 2017.
2. William Stallings, "Data and Computer Communications", Prentice Hall(PHI) , 8th /9th Edition, 2010/2011

References:

1. James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson Education,5th /7th edition, 2012/2016

Course Objectives: The objective of the course is to provide a foundation and clear understanding of various concepts of data communication which will form basis of computer networking. Objectives are further divided as:

1. To elaborate various features and operations of data communication.
2. To inculcate protocol functions and issues related to Data Link layer.
3. To introduce the design and configuration of various networking techniques.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	**Describe** fundamental concepts of data communication system.	2	Understanding
CO2	**Interpret** various concepts related to data link layer protocols.	3	Applying
CO3	**Differentiate** and **analyze** various data communication techniques	4	Analyzing

CO-PO Mapping :

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

1: Low, 2: Medium, 3: High

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 etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Introduction	**4 Hrs.**
A Communications Model, Data Communications, Networks, The Internet-An Example	

Configuration. Data communication Concepts and Terminology: Analog and Digital Data Transmission, Transmission Impairments, Channel Capacity. Media: Guided Transmission Media, Wireless Transmission, Wireless Propagation, Line-of-Sight Transmission Types of electronics communication, Electromagnetic spectrum, Bandwidth, Signal Types, Noise: internal, External, Noise calculation.	
Module 2: Encoding techniques	**8 Hrs.**
Digital Data- Digital Signals, Digital Data- Analog Signals, Analog Data- Digital Signals, Analog Data- Analog Signals. Digital data communication techniques:- Asynchronous and Synchronous Transmission, Types of Errors, Error Detection & Correction, Hamming Code, CRC, Checksum, Line Configurations, Numerical problems on encoding.	
Module 3: Multiplexing	**8 Hrs.**
Frequency Division Multiplexing, Synchronous Time Division Multiplexing, Statistical Time Division Multiplexing, Pulse code modulation, Delta modulation, Adaptive delta modulation, Differential PCM, PAM. Spread Spectrum: The Concept of Spread Spectrum, Frequency-Hopping Spread Spectrum, Direct Sequence Spread Spectrum, Code Division Multiple Access.	
Module 4: Switching techniques	**8 Hrs.**
Switched Communications Networks, Circuit-Switching Networks, Circuit-Switching Concepts, Soft switch Architecture, Packet-Switching Principles, X.25, and Frame Relay. Introduction to Asynchronous Transfer mode protocol Architecture, Logical Connections, ATM Cells, Routing in Arpanet.	
Module 5: Congestion control	**5 Hrs.**
Effects of Congestion, Congestion Control, Traffic Management, Frame Relay Congestion Control. Cellular wireless network: Principles of Cellular Networks, First-Generation Analog Second-Generation CDMA, Third-Generation Systems.	
Module 6: Flow Control and Internet Reference Models	**6 Hrs.**
Framing –Fixed, Variable error control, Flow control, Simplest Protocols, Stop & Wait Protocols, GO Back N & Selective Repeat Sliding window protocols, Numerical problems on flow control techniques, other Protocols. Internet and Reference models-OSI, TCP/IP.	

Module wise Measurable Students Learning Outcomes :

After the completion of the course the student should be able to:

Module 1 :

- Describe Data Communication model and various media for communication

Module 2 :

- Differentiate different encoding techniques.
- Apply error control techniques.

Module 3 :

- Differentiate and analyze various multiplexing techniques.

Module 4 :

- Distinguish between different switching techniques.

Module 5 :

- Identify and describe congestion control mechanisms and cellular wireless network.

Module 6 :

- Describe and differentiate various flow control techniques.

Title of the Course: Computer Organization and Architecture Course Code : **5CS205**	L	T	P	Cr
	3	0	0	3

Desirable requirements: Basic Electronics Engineering

Textbooks:

1. William Stallings, "Computer Organization and Architecture: Designing for Performance", Pearson Education, 8th Edition/10th Edition, 2010/2016
2. Ramesh S. Gaonkar, "Microprocessor architecture, programming & applications", Penram International publications (India) Pvt. Ltd, 6th edition, 2013
3. N. Senthil Kumar, M. Saravanan, S. Jeevanathan, S. K. Shah, "Microprocessors and Interfacing", Oxford Higher Education, 1st Edition, 2012

References:

1. David A. Patterson and John L. Hennessy "Computer Organization and Design: The Hardware/Software Interface", Elsevier, 5th Edition, 2013
2. Ram, "Fundamentals of Microprocessors and Microcontrollers", Dhanpat Rai Publications, 1st edition, 2012
3. ARM Based Development course, NPTEL(https://nptel.ac.in/courses/117106111/)

Course Objectives: The main objective of this course is to introduce and provide insights regarding different organizations and architectures of computer. The objectives are further divided as:

1. To introduce organization and architecture of computer.
2. To provide a foundation to write an 8 bit microprocessor program using assembly language.
3. To infuse understanding of usefulness X-86 microprocessor family and other processors and fundamental principles of ARM processors.

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	**Describe** basic concepts of the organization and architecture of computer and interfacing with external devices.	2	Understanding
CO2	**Illustrate** the knowledge gained about the data representation, internal organization, addressing modes, instruction set of 8085, 8086 and ARM processor for assembling language programming.	3	Applying
CO3	**Analyze** the working of processors like 8085,8086,ARM and interfacing of external devices like memory and I/O.	4	Analyzing

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**3**	**1**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**
CO2	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO3	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**

1: Low, 2: Medium, 3: High

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 etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Introduction to Computer Organization	**6Hrs.**
Introduction to Computer Organization and architecture, A brief history of computers, Von Neumann Architecture, designing for performance, Multicore, MICs and GPGPUs, Two Laws that Provide Insight: Amdahl's Law and Little's, Basic Measures of Computer Performance: Clock Speed, Instruction Execution Rate. Top level view of computer function and evolution: Computer Components, Computer Function, Interconnection Structures, Bus Interconnection, Point-to-Point Interconnect, PCI Express.	
Module 2: Data Representation and Computer Arithmetic	**6Hrs.**
The Arithmetic and Logic Unit, Integer Representation, Integer Arithmetic, Floating-Point Representation, Floating-Point Arithmetic, Programmable Logic Devices.	
Module 3: 8085 Microprocessor	**8Hrs.**
CPU organization, Microprocessors, Machine language, Assembly Language, Computer classification, Microprocessor Architecture, microcomputer systems; Single chip microcomputer: Microcontrollers, The 8085 microprocessor, machine cycles, 8085 Programming model, Instruction classification, Instruction Data format and storage, 8085 Instructions: Data transfer operations, Arithmetic operations, Logic operations, Branch operations.	
Module 4: X-86 microprocessor Family	**7Hrs.**
Microprocessor Architecture -8086, Register organization of 8086, Signal descriptions of 8086 chip, Physical Memory organization, Introduction to Maximum and Minimum mode operation, Addressing Modes, Co-processor configuration, interfacing of Co-processor with 8086.	
Module 5: Interfacing of Memory & Input / Output Devices	**7Hrs.**
Memory mapped I/o and I/O mapped I/O. Address decoding, interfacing of memory chips with 8085. Interfacing of interrupt controller with 8085, Programmable Interrupt Controller (8259A). Direct Memory Access (DMA), Stacks and subroutines.	
Module 6: Introduction to ARM Processor	**6Hrs.**
Arm core dataflow model, Registers, Current program status register, Pipeline, Exception, interrupt and vector table, Core extensions, Arm processor families, Data processing instruction and Arithmetic instruction.	

Module wise Measurable Students Learning Outcomes :
After the completion of the course the student should be able to:

Module 1:

- Describe different computer components.
- Illustrate Basic Measures of Computer Performance and its use.

Module 2:

- Illustrate computer arithmetic with examples.

Module 3:

- Describe the basics of microprocessors and analyze difference between the machine language and assembly language of a computer.
- Describe the peculiarities of the instructions as to their category, word size, machine cycles for execution, addressing mode etc.

Module 4:

- Illustrate the basics of X-86 family microprocessors and describe the functions of each of its components.

Module 5:

- Describe the interfacing between microprocessor and various peripherals.

Module 6:

- Illustrate ARM processor family using assembly instructions and their formats and usage.

Title of the Course: Software Engineering	L	T	P	Cr
Course Code: **5CS206**	3	0	0	3

Desired requirements:

Textbooks:

1. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa Publishers, 3rd Edition, 2005.
2. Ian Sommerville, "Software Engineering", Addison-Wesley, 7th Edition, 2004.
3. James Rumbaugh, "Object Oriented Modeling and Design with UML", Pearson, 2nd Edition, 2004.

References:

1. Roger S. Pressman, *"*Software Engineering: Practitioner's Approach*",* McGraw Hill, 7th Edition, 2010.
2. Jawadekar W.S., *"*Software Engineering: principles and practices*",* Tata McGraw Hills, 1st Edition.
3. Gillies A.C. and Smith p., *"*Managing Software Engineering: CASE studies and solutions*",* Chapman and Hall, London.

Course Objectives :

1. To unleash the orientation & importance of engineering approach to software development.
2. To infuse the knowledge of software processes & models practiced at IT industries.
3. To acquaint students with the SDLC phases in detail.
4. To emphasize on Design aspect with UML technology.
5. To inculcate the importance of software quality by virtue of software testing methods.

Course Learning Outcomes:

CO	**After the completion of the course the student Learner be able to**	Bloom's Cognitive level	Descriptor
CO1	**Grasp** industry processes on software development to become IT industry-savvy.	2	Understanding
CO2	**Prepare** with the spirit of team-working and importance of using artifacts at SDLC phases.	3	Applying
CO3	**Distinguish** and evaluate procedural & OO based development practices.	4	Analyzing
CO4	**Integrate** expertise on CASE tools usage especially for design and testing of software to undertake industrial strength software projects.	6	Creating

CO-PO Mapping :

PO and PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	-	-	3	-	-	-	-	-	-	-	3	2	3	-
CO2	-	-	1	2	-	-	-	3	3	3	-	-	-	-
CO3	-	-	-	-	2	-	-	-	-	-	-	-	-	-
CO4	-	-	2	-	-	-	-	-	-	-	-	2		3

1: Low, 2: Medium, 3: High

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 pedagogy such as brainstorming, role play, quiz, presentations etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Software Processes	**6 Hrs.**
Need of software engineering approach, ETVX model, project management process, software development process & models, configuration management process, process management process.	
Module 2: Software Quality & Project Planning	**6 Hrs.**
Quality objectives, software quality factors, PAF Model, quality standards, project management plan, cost estimation, project scheduling, personnel planning with WBS, risk management.	
Module 3: Software Requirement Analysis & Function Oriented Design	**7 Hrs.**
Software requirement process, need and characteristics of SRS artifact, design principles, module level concepts, design notation and specifications, structured	

design methodology.	
Module 4: Object Oriented Design with UML & Continual Integration	**8 Hrs.**
UML model, UML diagrams: Use-case, Class, Activity, State-chart, Interaction, Sequence, Collaboration, Component, Deployment. Continual integration with Agile model process frameworks.	
Module 5: User Interface Design & Coding	**4 Hrs.**
UI rules, UI analysis and steps in UI design, best programming practices such as TDD & pair programming, verification.	
Module 6: Software Testing	**8 Hrs.**
Testing purpose and concepts, test process, levels of testing, regression testing, test case design for functional testing & structural testing. Study of Open-source Tools.	

Module wise Measurable Students Learning Outcomes :

The student should be able to:

Module 1: Software Processes

- Awareness of Software processes & Models used at IT.

Module 2: Software Quality & Project Planning

- Grasp quality parameters and standards such as PAF.
- Know & prepare project planning phases and responsibilities with WBS.

Module 3: Software Requirement Analysis & Function Oriented Design

- As per SDLC phase understand requirement process and need of SRS artifact. Understand functional & non-functional requirements as well. Realize the importance of design aspects, concepts & methodology. Practices to learn how to draw DFD on requirements.

Module 4:Object Oriented Design with UML & Continual Integration

- Building capability to draw & distinguish various UML diagrams on requirements. Articulating usage of Continual integration with Agile model process frameworks.

Module 5: User Interface Design & Coding

- Know the UI aspect of interactive design for enterprise applications.
- Learn best coding standards/practices such as TDD, pair programming and how to verify code.

Module 6: Software Testing

- Integrate expertise on how testing helps in quality of software. Know testing concepts, levels of testing. Learn and practice Black & white box testing along with test case generations using open-source tools.

ODD Semester

Professional Core (Lab) Courses

Title of the Course: Programming Lab 1 **Course Code: 5CS251**	L	T	P	Cr
	0	0	2	1

Desirable Requirements: Introduction to any Programming Language

Textbooks:

1. Herbert Schildt, "The Complete Reference: C++*"* Tata McGraw-Hill, 4th Edition, 2010
2. E Balaguruswamy, "Object Oriented Programming with C++*",* Tata McGraw-Hill, 4th Edition, 2008
3 .Kenneth Lambert, "Fundamentals of Python: First Programs*"* Course Technology, Cengage Learning.2nd edition, 2017

References:

1. Stanley B. Lippman , "C++ Primer" Pearson , 4th Edition, Jan 2010

Course Objectives :

The course covers fundamentals of object oriented concepts using C++ & Python programming with syntax and examples. Course objectives of this course are as follows:

1. To provide in-depth coverage of object-oriented programming principles and techniques using C++ and Python.
2. To inculcate the advanced programming concepts in C++ and Python.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** the features of object oriented programming using C++ and Python.	2	Understanding
CO2	**Demonstrate** the solution to real world problems using C++ and Python	3	Applying

CO-PO Mapping :

PO and PSO	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	-	-	-	-	-	-	-	-	-	-	-	-	**1**	-
CO2	-	-	-	-	**2**	-	-	-	-	-	-	-	**2**	-

1:Low, 2:Medium, 3:High

Assessments:

Lab Assessment:

There are four components of lab assessment, LA1, LA2, LA3 and Lab ESE.
IMP: Lab ESE is a separate head of passing.

Assessment	Based on	Conducted by	Conduction and Marks Submission	Marks
LA1	Lab activities, attendance, journal	Lab Course Faculty	During Week 1 to Week 4 Submission at the end of Week 5	25
LA2	Lab activities, attendance, journal	Lab Course Faculty	During Week 5 to Week 8 Submission at the end of Week 9	25
LA3	Lab activities, attendance, journal	Lab Course Faculty	During Week 10 to Week 14 Submission at the end of Week 14	25
Lab ESE	Lab Performance and related documentation	Lab Course faculty	During Week 15 to Week 18 Submission at the end of Week 18	25

Week 1 indicates starting week of Semester.

Lab activities/Lab performance shall include performing experiments, mini-project, presentations, drawings, programming and other suitable activities, as per the nature and requirement of the lab course.
The experimental lab shall have typically 8-10 experiments.

Course Contents:

Module 1: Introduction to object oriented programming	**5 Hrs.**
Introduction to properties of object oriented programming, Beginning with c++ programs, operators, control structures, loops, examples with class and objects, Functions in c++, function overloading, Constructors, Destructors, operator overloading, static class members. **Experiments:** 1. Program based on creating Class and Object. 2. Program based on constructor and destructor.	
Module 2: Properties of object oriented programming	**6 Hrs.**
Inheritance and its types, pointers, virtual functions, Polymorphism. File Handling, Exception Handling, Templates, and Namespace fundamentals. Overview of Stream classes **Experiments:** 1. Implementation of Inheritance and polymorphism. 2. Working with files. 3. Use of template, generic template and function. 4. Creation of namespaces.	
Module 3: Introduction and getting started with python programming	**4 Hrs.**
Running Code in the Interactive Shell, Input, Processing, and Output, Editing, Saving, and Running a Script, Behind the Scenes: How Python Works. Data Types and expressions: Numeric Data Types and Character Sets, Integer, Floating-Point Numbers, Character Sets, Arithmetic Expressions, Functions and Modules. **Experiments:** 1. Introduction and getting started with python programming: running code in the interactive shell 2. Program based on expression, data type, functions	
Module 4: Features of python programming	**4 Hrs.**
Loops and selection statements, String, Lists and dictionaries: List Literals and Basic Operators, List Methods for Inserting and Removing Elements, Dictionary Literals. **Experiments:** Programs based on implementation of loops, strings, lists and dictionaries.	
Module 5: Design with Classes	**4 Hrs.**
Getting Inside Objects and Classes, A First Example: The Student Class, Graphical User Interfaces, Coding Simple GUI-Based Programs, Windows GUI components. **Experiments:** Programs based on Graphical user interface design using python.	
Module 6: Multi-Threading, Exception handling and File handling using python	**3 Hrs.**
Multi-threading, Exception handling, file handling. **Experiments:** 1. Programs related to Multi-threading, Exception handling, file handling.	

Title of the Course: Data Structures Lab **Course Code: 5CS253**	L	T	P	Cr
	0	0	2	1

Desirable requirements: Programming in C including pointers and File Handling

Textbooks:

1. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures, A Pseudocode Approach With C", Cengage Learning, Second Edition, 2014
2. S. Lipschutz, "Data Structures", Schaum's Outlines Series, Tata McGraw-Hill, 2013
3. Ellis Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi, 2008

References:

1. Yashavant Kanetkar, "Understanding pointers in C", BPB Publication, 2009
2. N. B. Venkateshwarlu, E. V. Prasad, "C and Data Structures", S. Chand and Company, 2010

Course Objectives: This laboratory course focuses on practicing various linear and nonlinear data structures introduced in the Data Structure theory course. The assignment list mainly emphases on explaining characteristics of various data structures and their applications.

1. To develop and improve skills in programming in a systematic way and preparing the students for advanced computer science courses.
2. To make the students understand the concept of ADT, recursion, various searching and sorting algorithms along with their performance comparisons and to use appropriate data structure for modelling given problem.
3. To inculcate theoretical and practical knowledge of various linear and nonlinear data structures to solve real world problems.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Demonstrate** the concept of recursion, abstract properties of various linear and nonlinear data structures, searching and sorting methods through implementation.	3	Applying
CO2	**Identify** suitable data structure to be used to solve the various problems.	4	Analyzing
CO3	**Select** appropriate searching, sorting method on the basis of its performance while developing application.	5	Evaluating

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	-	-	-	-	**2**	-	-	-	-	-	-	-		-
CO2	-	-	-	**3**	**2**	-	-	-	-	-	-	-	**3**	-
CO3	-	-	-	**3**	**2**	-	-	-	-	-	-	-	**3**	-

1: Low, 2: Medium, 3: High

Assessments :
Lab Assessment:

There are four components of lab assessment, LA1, LA2, LA3 and Lab ESE.
IMP: Lab ESE is a separate head of passing.

Assessment	Based on	Conducted by	Conduction and Marks Submission	Marks
LA1	Lab activities, attendance, journal	Lab Course Faculty	During Week 1 to Week 4 Submission at the end of Week 5	25
LA2	Lab activities, attendance, journal	Lab Course Faculty	During Week 5 to Week 8 Submission at the end of Week 9	25
LA3	Lab activities, attendance, journal	Lab Course Faculty	During Week 10 to Week 14 Submission at the end of Week 14	25
Lab ESE	Lab Performance and related documentation	Lab Course faculty	During Week 15 to Week 18 Submission at the end of Week 18	25

Week 1 indicates starting week of Semester.
Lab activities/Lab performance shall include performing experiments, mini-project, presentations, drawings, programming and other suitable activities, as per the nature and requirement of the lab course.
The experimental lab shall have typically 8-10 experiments.

Course Contents:

List of Experiments:

Assignments based on topics covered in course 4CS203

1. Program based on structures and pointers in C
2. Program based on arrays and pointers in C
3. File handling and command line arguments
4. Implementation of recursion
5. Developing ADT for singly linked list and its applications
6. Developing ADT for Doubly linked list and its applications
7. Developing ADT for circular linked list and its applications
8. Developing ADT for stack and queue and their applications
9. Implementation of double ended queue
10. Implementation of recursive and non-recursive tree traversals
11. Binary search tree and application
12. Implementation of graph, DFS, BFS
13. Implementation of searching : linear search, binary search, Fibonacci search
14. Sorting Methods: Insertion sort, shell sort, heap sort, quick sort, merge sort, radix sort etc.
15. Implementation of hashing

Title of the Course: Computer Organization And Architecture Lab **Course Code**: **5CS255**	L	T	P	Cr
	0	0	2	1

Desirable requirements: Programming by using assembly language

Textbooks:

1. William Stallings, "Computer Organization and Architecture: Designing for Performance", Pearson Education, 8th Edition/10th Edition, 2010/2016
2. Ramesh S. Gaonkar, "Microprocessor architecture, programming & applications", Penram International publications (India) Pvt. Ltd, 6th edition, 2013
3. N. Senthil Kumar, M. Saravanan, S. Jeevanathan, S. K. Shah, "Microprocessors and Interfacing", Oxford Higher Education, 1st Edition, 2012

References:

1. David A. Patterson and John L. Hennessy "Computer Organization and Design: The Hardware/Software Interface", Elsevier, 5th Edition, 2013
2. Ram, "Fundamentals of Microprocessors and Microcontrollers", Dhanpat Rai Publications, 2012
3. ARM Based Development course, NPTEL(https://nptel.ac.in/courses/117106111/)

Course Objectives:

The main objective of this course is to demonstrate insights regarding working of different organizations, components and architectures of computer. The objectives are further divided as:

1. To infuse skills of drawing flowchart by using assembly language programming.
2. To demonstrate block transfer, arithmetical, logical operations and code conversion method by using assembly language programs.
3. To demonstrate the working of ARM processor.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Grasp** the fundamentals of assembly level programming using microprocessor trainer kit and interfacing with other I/O devices.	2	Understanding
CO2	**Demonstrate** programming proficiency using the various addressing modes and instructions set (Block transfer, arithmetical, logical operations and code conversion method) of 8085 and X-86 microprocessor.	3	Applying

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**2**	**-**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO2	**-**	**-**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

1: Low, 2: Medium, 3: High

Assessments :

Lab Assessment:

There are four components of lab assessment, LA1, LA2, LA3 and Lab ESE.

IMP: Lab ESE is a separate head of passing.

Assessment	Based on	Conducted by	Conduction and Marks Submission	Marks
LA1	Lab activities, attendance, journal	Lab Course Faculty	During Week 1 to Week 4 Submission at the end of Week 5	25
LA2	Lab activities, attendance, journal	Lab Course Faculty	During Week 5 to Week 8 Submission at the end of Week 9	25
LA3	Lab activities, attendance, journal	Lab Course Faculty	During Week 10 to Week 14 Submission at the end of Week 14	25
Lab ESE	Lab Performance and related documentation	Lab Course faculty	During Week 15 to Week 18 Submission at the end of Week 18	25

Week 1 indicates starting week of Semester.
Lab activities/Lab performance shall include performing experiments, mini-project, presentations, drawings, programming and other suitable activities, as per the nature and requirement of the lab course. The experimental lab shall have typically 8-10 experiments.

Course Contents:

Assignments based on topics covered in course 4CS252

Write assembly language programs

1. Introduction to digital fundamental circuit design.
2. Study of the design combinational and sequential circuit.
3. Introduction of Microprocessors and Study of 8085 Microprocessor and instruction set.
4. Write a program to perform 8-bit block transfer.
5. Write a program to perform 8-bit and 16-bit addition /subtraction/multiplication/division.
6. Write a program to find largest /smallest number in an array of data.
7. Write a program to find smallest no in an array of data.
8. Write a program to find 16 bit 2's complement no of 4340H
9. Write a program to transfer 16 bytes of data stored in location at C250 to C25F to new memory locations starting from C300 on words.
10. Write a program to transfer a block of data. The data is stored in memory from C550 H to C555F H. The data is to be stored from C570 H to C57F H in reverse order.
11. Write a program to arrange 10 bytes data in ascending /descending order. The data is stored in memory as an array starting from C100 H onwards.
12. Write Convert a binary number to a BCD number.
13. Write a program to square of number using lookup table.
14. Write X86/64 ALP to perform basic arithmetic operation.
15. Write X86/64 ALP to count number of positive and negative numbers from the array.
16. 16. Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method (Use of 64-bit registers is expected).
17. Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number.
18. Case study: ARM Processor.

ODD Semester

Minor Specialization Courses

Title of the Course: Data Structures Course code: 1CSM01	L	T	P	Cr
	3	0	0	3

Pre-Requisite Courses: Programming in C including pointers and File Handling

Textbooks:

1. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures, A Pseudocode Approach With C",Cengage Learning, Second Edition, 2014
2. S. Lipschutz, "Data Structures,Schaum's" Outlines Series, Tata McGraw-Hill, 2013
3. Ellis Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++*",* Galgotia Book Source, New Delhi, 2008

References:

1. Yashavant Kanetkar, "Understanding pointers in C*",* BPB Publication, 4th Edition, 2009
2. Jean-Paul Tremblay, Paul. G. Soresan, "An introduction to data structures with Applications*",* Tata Mc-Graw Hill International Editions, 2nd edition 1984
3. N. B. Venkateshwarlu, E. V. Prasad, "C and Data Structures*",* S. Chand and Company, 2010

Course Objectives :

1. To impart basic concepts of Data Structure and analyse efficiency of algorithm
2. To make the students understand elementary linear and non-linear data structures and make the students capable of applying appropriate data structure for modelling a given problem.
3. To provide a foundation to analyse and apply various searching and sorting techniques.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1	Explain the fundamental concepts of various linear and non-linear data structures with ADTs and write recursive algorithms.	2	Understanding
CO2	Identify suitable data structures to be used and apply it to solve the various problems.	3	Applying
CO3	Compare and analyse various searching and sorting methods based on inherent properties of data structures and the complexity of algorithms.	4	Analyzing

CO-PO Mapping :

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

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 etc.

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

ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1 Introduction **Basic Concepts:** Algorithm, Pseudocode, ADT, Data Structure, Algorithmic Efficiency **Recursion:** Direct and Indirect recursion, Recursive solution of Towers of Hanoi.	**Hrs. 5**
Module 2 Linked Lists Concept of linked organization, Singly linked list, doubly linked list and dynamic storage management, circular linked list, Operations such as traversal, Searching, insertion and deletion, Representation of polynomials using linked lists	**Hrs. 7**
Module 3 Stacks and Queues Fundamentals stack and queue as ADT, Representation and Implementation of stack and queue using sequential and linked organization, Insert and Delete operations, Circular queue: representation and implementation, Priority queue	**Hrs. 8**
Module 4 Trees Basic terminology, binary trees and its representation, binary tree traversals, AVL Tree, Binary Search Trees, Heaps and its operations	**Hrs. 7**
Module 5 Graphs Terminology and Representation of graphs using adjacency matrix, adjacency list and adjacency matrix, Traversals Depth First and Breadth First, Minimum Spanning Tree	**Hrs. 5**
Module 6 Searching & Sorting Technique **Search:** Importance of searching, Sequential, Binary, Fibonacci search **Sorting:** Importance of sorting, Insertion, Heap, Quick sort, Merge sort **Hashing: Introduction** Hash functions.	**Hrs. 7**

Module wise Measurable Students Learning Outcomes :

After the completion of the course the student should be able to:

Module 1:

- Explain the fundamental concepts and write recursive algorithms.

Module 2:

- Demonstrate concept of linked list and use of ADTs to solve the problem

Module 3:

- Discuss theoretically and use data structures like stacks and queues as the programmers' tool to develop the solution.

Module 4:

- Explain non-linear data structure tree and its basic operations.

Module 5:

- Discuss and implement graphs using various representations.

Module 6:

- Explain and compare various searching and sorting techniques.

EVEN Semester

EVEN Semester

Professional Core (Theory) Courses

Title of the Course: Applied Mathematics for Computer Science and Engineering	L	T	P	Cr
Course Code: 5CS221	3	0	0	3

Desirable requirements: Engineering Mathematics I and Engineering Mathematics II

Textbooks:

Textbooks:

1. Gilbert Strang, "Linear Algebra and its applications", Cengage Learning, 4th edition, 2014
2. George J. Klir and Bo Yuan, " Fuzzy Sets and Fuzzy Logic : Theory and Applications", Pearson Education Services Pvt. Ltd., 4th edition, 2017
3. Timothy C. Urdan, " Statistics in Plain English", Routledge-Taylor and Fransis Group, 3rd Edition, Volume 1, 2010.
4. Alice Zheng, "Evaluating Machine Learning Models" O'Reilly Media, 2015

References:

2. Seymour Lipschutz and Mark Lipson,"Schaum's outlines of Theory and Problems of Linear Algebra", Tata McGraw Hill, 3rd Edition, 2007.
3. William Stein, "Elementary Number Theory: Primes, Congruences, and Secrets", Springer, 1st Edition, 2008.

Course Objectives: The main objective of this course is to build a foundation for solving problems in different domains such as Machine Learning, Artificial Intelligence, Scientific Computing, Cryptography etc. The objectives are further divided as:

2. To infuse an understanding of the mathematical theory of Linear Algebra, Evaluation metrics for computer science engineers.
3. To provide a foundation to solve practical problems in cryptography, data science and machine learning.
4. To give insights about the properties, operations and relations on Fuzzy sets.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Illustrate** the concept of Linear Algebra and Fuzzy sets with case studies.	2	Understanding
CO2	**Apply** various evaluation metrics for result analysis	3	Applying
CO3	**Solve** mathematical problems using tools from mathematical areas, including algebra, analysis, evaluation metrics and number theory.	3	Applying

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO2	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**
CO3	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**

1: Low, 2: Medium, 3: High

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/oral etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1 Vector Spaces	**6 Hrs.**
Introduction, Vector spaces, Linear combinations, Spanning sets, Subspace, Linear dependence and independence, Basis and dimension, Null space, Column space, Row space, Rank-Nullity theorem.	
Module 2 Advanced Concepts in Linear Algebra	**7 Hrs.**
Vector dot product, Inner product space, Length and orthogonality, Orthogonal sets, Orthonormal sets, Orthogonal projections, Gram-Schmidt Process, Least square problems, Applications and significance of Eigen values and Eigen vectors.	
Module 3 Fuzzy Sets	**7 Hrs.**
Introduction to characteristics functions, First decomposition theorem, Fuzzy relations, examples, Fuzzy equations, Operations on Fuzzy sets.	
Module 4 Exploratory Data Analysis	**6 Hrs.**
Discrete and continuous random variables, PDF, CDF, percentile, Inter quartile range, central tendency (mean, mod, median, dispersion, skewness, kurtosis), variance, standard deviation, Mean Absolute Deviation (MAD), Standardization (Z-score), Normalization.	
Module 5 Evaluation metrics	**6 Hrs.**
Intersection over union (IoU), Inception score, Frechet Inception distance, BLEU, METEROR, Rough, CIDER score, Confusion Matrix, F1 Score, Recall or Sensitivity, Gain and Lift Charts, Kolmogorov Smirnov Chart, AUC – ROC, Log Loss, Gini Coefficient, Concordant – Discordant Ratio, Root Mean Squared Error.	
Module 6 Number theory	**7 Hrs.**
Primality Testing: Primality Tests, Pseudo primes, Fermat's pseudo primes, Factorization techniques, Multiplicative inverse. Euclidean algorithm, Chinese remainder theorem, Fermat's little theorem, Wilson's theorem, Primitive roots, Quadratic residues.	

Module wise Measurable Students Learning Outcomes :
After the completion of the course the student should be able to:

Module 1:
- Solve the problems on vector spaces.

Module 2:
- Apply the core concept of Linear Algebra to real life applications.

Module 3:
- Illustrate the knowledge of fuzzy set and system through set theory**.**

Module 4:
- Visualize and understand the different data from science and engineering domains using statistics tools.

Module 5:
- Make use of appropriate evaluation mentics in the analysis of simple datasets or proposed method.

Module 6:
- Identify how number theory is related to and used in cryptography.

Title of the Course: Formal Language and Automata Theory **Course code: 5CS222**	L	T	P	Cr
	3	1	0	4

Desirable requirements: Discrete Mathematics

Textbooks:

1. John C. Martin, "Introduction to Languages & Theory of Computation", Tata McGraw-Hill , 3rd Ed., 2009
2. John E.Hopcraft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages and Computations", Pearson Edu., 3rd Ed., 2009
3. Daniel I. A. Cohen, "Introduction to Computer Theory", Wiley, 2nd Ed., 2008

References:

1. J.P.Tremblay & R.Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw-Hill, 2008
2. K.L.P. Mishra & N. Chandrasekaran, "Theory of Computer Science", PHI, 2nd Ed., 2002
3. Vivek Kulkarni, "Theory of Computation", Oxford University Press, 1st Ed., 2013

Course Objectives :

This course is one of the core subjects for Computer Science and Engineering students which deals with the theory related to the practical aspects of computation. The main emphasis is on solving problems universally encountered in designing a language translator, regardless of source or target machine.

1. To explain basic terminologies related to formal languages and Automata theory.
2. To provide foundation to critically analyze grammars, regular expressions, languages, and their relationship.
3. To inculcate theoretical knowledge to design Automata/Machine as a language descriptor and recognizer.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Explain** the fundamental concepts related to string, language, grammar and their properties	2	Understanding
CO2	**Examine** and **Construct** different grammars, regular expressions and relate the languages defined by different grammars and regular expressions.	3	Applying
CO3	**Design** Finite Automata, PDA, Turing Machine to recognize different languages.	6	Creating

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO2	**3**	**3**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**
CO3	**3**	**3**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**

1: Low, 2: Medium, 3: High

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
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1	**6 Hrs.**
Types of Proofs, Mathematical Induction and Recursive definitions, Regular expressions & corresponding regular languages, examples and its applications, unions, intersection & complements of RL, Pumping Lemma for RL.	
Module 2	**10 Hrs.**
Deterministic finite automata definition and representation, Nondeterministic F.A., NFA with ^ transitions, Equivalence of DFAs, NFAs and NFA-^s. Kleene's theorem & proofs, minimum state FA for a regular language, minimizing number of states in an FA.	
Module 3	**6 Hrs.**
Definition and types of grammars and languages, derivation trees and ambiguity, CFL's & Non CFL's., Union, Concatenation and Kleene's operations, Intersection and complements of CFLs, Pumping Lemma & examples.	
Module 4	**6 Hrs.**
Definition, deterministic PDA, types of acceptance and conversions to each other, CFGs & PDAs, Top-Down, & Bottom-up parsing.	
Module 5	**4 Hrs.**
BNF, CNF and GNF notations, Eliminating ^ production and unit productions from a CFG, Eliminating useless variables from a Context Free Grammar.	
Module 6	**7 Hrs.**
Models of computation, definition of TM as Language Acceptors, Combining Turing Machines, computing a function with a TM. Variations in TM, TMs with doubly-infinite tapes, more than one tape, Nondeterministic TM and Universal TM.	

Module wise Measurable Students Learning Outcomes :
After the completion of the course the student should be able to:

Module 1:
- Explain basic terminologies related to theory of computation and construct regular expressions recognising regular languages.

Module 2:
- Explain finite state system and design finite automata for regular languages.

Module 3:
- Explain language syntax, grammar construct context free grammars for languages.

Module 4:
- Design push down automata and demonstrate different parsing techniques.

Module 5:
- Explain different normal forms and their applications.

Module 6:
- Design Turing machines for different formal languages and illustrate variants of Turing machine.

Tutorial:
Based on the syllabus, 15 assignments will be given to the students focusing on problem solving approach.

Title of the Course: Operating Systems **Course Code: 5CS223**	L	T	P	Cr
	3	0	0	3

Desirable requirements:

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", John Wiley, 10th Edition, 2018
2. D. M. Dhamdhere,"Operating Systems A Concept-Based Approach", McGraw-Hill, 3rd edition, 2012

References:

1. Charles Crowley, "Operating System A Design Oriented Approach", McGraw-Hill Education Pvt. Ltd., 2001
2. Achyut S. Godbole, Atul Kahate "Operating System with Case Studies in Unix, Netware and Windows NT", Tata McGraw Hill,3rd edition, 2010
3. D.M.Dhamdhere, "System Programming and Operating Systems", Tata McGraw - Hill, 2nd Edition, 1999

Course Objectives: A successful student will be able to understand the basic components of a computer operating system, and the interactions among the various components. The course will cover an introduction on the policies for scheduling, deadlocks, memory management, synchronization, system calls, and file systems.

1. To introduce students with basic concepts of operating system, system software, threads and their communication
2. To familiarize the students with various views and management policies adopted by O.S. as pertaining with processes , Deadlock , memory , File and I/O operations.
3. To provide the knowledge of basic concepts towards process synchronization, Mutual exclusion algorithms and deadlock detection algorithms and related issues.
4. To inculcate importance of memory management, storage management and I/O device management in OS design.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Describe** the primitive concepts of Operating System services and system software functionality.	2	understanding
CO2	**Illustrate** Process management, Memory management, Storage management and I/O management core techniques in effective execution of processes.	3	applying
CO3	**Assess** various algorithms of Process, Memory, Storage & I/O management for performance and quality criterion.	5	evaluating

CO-PO Mapping :

PO and PSO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO 11**	**PO12**	**PSO1**	**PSO2**
CO1	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO2	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**-**
CO3	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**-**

1: Low, 2: Medium, 3: High

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 etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1 : Overview of Operating System	**6Hrs.**
Notion of operating systems ,Operating system services, user operating system interface, system calls, types of windows and UNIX system calls, system programs, operating system design and implementation, operating system structure, Virtual Machines **Case Study** : Windows and UNIX Operating System	
Module 2 : System Softwares	**6Hrs.**
Notions of editors, Macro processors, Compilers, Assemblers, loaders & linkers, Multiprogramming and time sharing.	
Module 3 : Process Management	**7Hrs.**
Process Concept : Process concept, process scheduling, operation on process, inter-process communication, example of IPC systems and communication in client-server systems. **Process Scheduling:** Basic concepts, scheduling criteria, scheduling algorithm, algorithm evaluation.	
Module 4 : Process Coordination	**7Hrs.**
Synchronization : Background, the critical section problem, Peterson's solution, synchronization hardware, semaphores, classic problems of Synchronization. **Deadlock :** System model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection.	
Module 5 : Memory Management	**8Hrs.**
Memory-Management Strategies : Background, swapping, contiguous memory allocation, paging, structure of the page table, Segmentation. **Virtual Memory Management :** Background, demand paging, copy-on-write, page replacement algorithms, allocation of frames, Thrashing.	
Module 6 : Storage Management	**5Hrs.**
File System : File concept, access methods, directory and disk structure, file-system mounting, file sharing, protection.	

Module wise Measurable Students Learning Outcomes :

Module 1:
- Describe the basic functions of OS, its components and its working.

Module 2:
- Illustrate the basic terminologies of system softwares like compiler, assembler.

Module 3:
- Describe the issues related to process management and solve the CPU scheduling problems.

Module 4:
- Describe different ways of process synchronization and handling deadlocks.

Module 5:
- Explain the issues related to memory management.

Module 6:
- Describe different ways of how logical view of File system is provided to the users by OS.

Title of the Course: Database Engineering **Course Code: 5CS224**	L	T	P	Cr
	3	0	0	3

Desirable requirements: Data Structures

Textbooks:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Mc-Graw Hill New York Publications, 6th Edition, 2011

References:

1. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", Mc-Graw Hill New York Publications, 3rd Edition, 2003
2. Ramez Elmasri and Shamkant Navathe, Benjamin Cummings, "Fundamentals of Database Systems", 3rd Edition, 1999 / later
3. Bipin c. Desai "An Introduction to Database System", Galgotia Publications, 2nd revised edition

Course Objectives : Goal of this course is to elucidate basics of database design, provide bird view of data using ER- model, introduce relation model to depict relation between them, discussion of clause to manipulate, access data using Query language, apply normalization to remove redundancy, explain access control mechanism for security and introduce storage and indexing strategies. Course objectives are as follows,

1. To Impart various functional components of database design, manipulation and access language, redundancy issue, storage strategy, transaction and concurrency strategy and its security and recovery system
2. To Introduce an physical and logical database designs, database modeling, relational, hierarchical and network models
3. To Provide in depth understanding of relational model and the theoretical issues associated with relational database design.
4. To Exemplify various SQL clauses of Data manipulation, Data access and Data control.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive level	Descriptor
CO1	**Explain** concepts of conceptual database design, redundancy problem, storage system, transaction processing, concurrency control and security in DBMS	2	Understanding
CO2	**Apply** theoretical knowledge to design ER diagram, prepare relational schema using appropriate constraints and normalization for a given specification of the requirement	3	Applying
CO3	**Construct** SQL queries for Open source and Commercial DBMS for a given specification schema to fetch essential data	3	Applying

CO-PO Mapping :

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

1: Low, 2: Medium, 3: High

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 etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Introduction and Database Modelling using ER Model	**6 Hrs.**
Introduction*:* General introduction to database systems, its advantages and applications, Database System Architecture, Database users and Administrator, Data models, Database management system, Database languages, View of Database, Data Models. **ER Model*:*** Entity set, Entity types, attributes, Notations, Relationship sets, Relationship types, Keys- super key, candidate key, primary key, Extended Features of ER Model-Generalization, Specialization and aggregation	
Module 2: Relational Model and SQL	**8 Hrs.**
Relational Model: Structure of Relational Database, Reduction of ER model into Relational schemas, Schema-instance distinction, Referential integrity and foreign keys, Relational algebra, Tuple relation calculus, Domain relational calculus, Example queries, **SQL**: Introduction to SQL, Data definition statements with constraints, Insert, Update and Delete, Set Operations, Aggregate functions group by and having clauses, Nested Queries, Views, Complex Queries, Joins.	
Module 3: Relational Database Design	**7 Hrs.**
Importance of a good schema design, Motivation for normal forms, Atomic domains and 1NF, Dependency theory - functional dependencies, Closure of a set of FD's, Definitions of 2NF, 3NF and BCNF, Decomposition algorithms and desirable properties of them, Multi-valued dependencies and 4NF, Join dependencies and definition of 5NF, Temporal Functional Dependencies	
Module 4: Data Storage and Indexing	**6 Hrs.**
File organization, Organization of records in files, Data Dictionary, Database Buffer, and Indexing: Concept, Ordered Indices-Primary, Secondary, Multilevel, B+ Tree Index, Hashing, Hash Indices, Dynamic hashing, Multiple key access, Bitmap Indices.	
Module 5: Transaction Processing and Concurrency Control	**7 Hrs.**
Transaction Processing: Concept, ACID properties, Transaction states, Storage Structure, Implementation of atomicity, isolation and durability, Serializability, Testing for serializability. **Concurrency Control**: Lock-based protocols, Timestamp - based Protocols, Validation - based Protocols, Multiple Granularities, Deadlock handling.	
Module 6: Database security and Recovery System	**5 Hrs.**
Authentication, Authorization and access control, Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role of the Database Administrator (RBAC) models, Intrusion detection, SQL injection. Failure classification, Recovery and Atomicity, Log based recovery, Checkpoints, Shadow Paging, Buffer management in crash recovery.	

Module wise Outcomes

At end of each module students will be able to

Module 1

- Explain the concept of database system and its applications, database system architecture and various database models, appropriate constraints, describes entity, attribute, relation, key, primary key, super key, candidate key, extended features specialization and generalization with notations.

Module 2

- Construction of ER- model, relation model to represent complex data in pictorial form for better design and extract information from the database by constructing SQL queries.

Module 3

- Illustrate and use the concept of functional dependency and various normal forms for "good" database design.

Module 4

- Describe file organization concepts and various indexing techniques.

Module 5

- Describe the concept of transaction and implement transactions and compare various concurrency control mechanisms and apply the concepts for hands-on experimentation.

Module 6

- Identify and define Authentication, Authorization and access control mechanisms for data security and recovery mechanisms using log based recovery, checkpoints, shadow paging, and Buffer management in crash recovery.

Title of the Course: Computer Network **Course Code: 5CS225**	L	T	P	Cr
	3	0	0	3

Desirable Requirements : Data Communication

Textbooks:

1. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw-Hill, 4th/5th edition, 2017
2. William Stallings, "Data and Computer Communications", Prentice Hall (PHI) , 8th/9th edition, 2010/2011
3. Andrew S. Tanenbaum, "Computer Networks", Prentice Hall (PHI), 3rd /5th Edition, 2008/2010

References:

1. James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson Education,5th /6th edition, 2012/2013
2. Thomas G. Robertazzi , "Computer Networks and Systems: Queueing Theory and Performance Evaluation", Springer, 2nd edition, 2000

Course Objectives: The course is designed to give a clear view of computer networking by introducing them to networking protocols, features and techniques. Objectives are divided as below:

1. To recall protocol functions and issues related to Data Link layer.
2. To explain the features and operations of various protocols in TCP/IP suite.
3. To elaborate the design and configuration of various networking protocols.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Articulate** networking basics and different layers in networking models	2	Understanding
CO2	**Examine** the features and operations of protocols of data Link Layer, Network layer, transport layer and Application Layer.	3	Applying
CO3	**Categorize** and **compare** networking protocols.	4	Analyzing

CO-PO Mapping :

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

1: Low, 2: Medium, 3: High

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 etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Networking Basics	**4 Hrs.**
Evolution of network, Introduction to Computer Networks, Types of Network, Physical & Logical Topology, Introduction to different types of network, internetworking, Intranet, Internet and revisit to Reference models-OSI, TCP/IP.	
Module 2:Data Link Layer	**8 Hrs.**
The Channel Allocation Problem-Static and Dynamic Allocation, Multiple Access Protocols-ALOHA, CSMA, CSMA/CD, WDMA, WLAN. Ethernet-cabling, coding, MAC Protocol, Binary exponential back off algorithm, performance, switched Ethernet, fast Ethernet, gigabit Ethernet. Wireless LANs-802.11 stack, physical layer, MAC, frame structure Bluetooth-architecture, application, protocol stack, Data Link Layer Switching- Bridge, hub, repeater, switch, router, gateways, VLAN.	
Module 3: The Network Layer:	**7 Hrs.**
Logical Addressing: IPv4 addresses , IPv6 addresses, internetworking, IPv4, IPv6, transition from IPv4 to IPv6, Address Mapping, ICMP, IGMP, Unicast and Multicast Routing, Numerical problems on logical addressing.	
Module 4: The Transport Layer:	**7 Hrs.**
Process-to-process delivery, user datagram protocol (UDP), TCP, SCTP, Socket Programming,	
Module 5: Congestion Control and Quality of Service	**6 Hrs.**
Congestion, congestion control, congestion control in TCP, introduction to queuing theory, quality of service, techniques to improve qos, integrated services, differentiated services.	
Module 6: Application Layer	**7 Hrs.**
Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP.	

Module wise Measurable Students Learning Outcomes :
After the completion of the course the student should be able to:

Module1 :

- Articulate the networking Basics.

Module2 :

- Explain and examine wired and wireless communication with medium access control layer.

Module3 :

- Understand the working of network layer and compare the techniques for routing at network layer.

Module4 :

- Examine the services provided by transport layer.

Module5 :

- Compare techniques to improve QoS. and congestion control

Module6 :

- Articulate knowledge of various application layer protocols.

EVEN Semester

Professional Core (Lab) Courses

Title of the Course: Database Engineering Lab Course Code: 5CS274	L 0	T 0	P 2	Cr 1

Desirable requirements: Data Structures

Textbooks:

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Mc-Graw Hill New York Publications, 6th Edition, 2011

References:

1. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", Mc-Graw Hill New York Publications, 3rd Edition, 2003
2. Ramez Elmasri and Shamkant Navathe, Benjamin Cummings, "Fundamentals of Database Systems", 3rd Edition, 1999 / later
3. Bipin c. Desai "An Introduction to Database System", Galgotia Publications, 2nd revised edition

Course Objectives : Main objective of this course is to practically demonstrate the ER- model for given specific requirement using open source or commercialized tool, show transformation of ER-model into Relation model on paper as well as on s/w tool, depict Relation model in table format on Open source or commercialized DBMS using query language and introduce advance topics to interact with DBMS like view, trigger, procedures and aspects of authorization.

1. To elaborate use of conceptual database designs to prepare database schemas, indexing, transaction processing, concurrency and recovery control issues associated with database management systems.
2. To make the students aware of various relational database systems and the systematic approach to apply theoretical knowledge to design practical applications to solve real world problems on the small scale.
3. To make the students understand SQL and to use it efficiently to retrieve data from the database.

Course Learning Outcomes:

CO	After the completion of the course the student should be able to	Bloom's Cognitive	
		level	Descriptor
CO1	**Interpret** the problem statement of an enterprise, identify the need, analyse the problem and design ER diagram for the enterprise as well as prepare the relational database schema for the enterprise identifying integrity constraints for efficient design using modern tools.	3	Applying
CO2	**Apply** systematically theoretical knowledge to design practical applications to solve real world database problems on the small scale and theoretically justify the design and use fundamental transaction processing, concurrency control etc. in real applications.	3	Applying
CO3	**Compare** and use various ways of writing the queries for a given problem and extract required information from the database.	4	Analyzing

CO-PO Mapping :

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

1: Low, 2: Medium, 3: High

Assessments :

Lab Assessment:

There are four components of lab assessment, LA1, LA2, LA3 and Lab ESE.
IMP: Lab ESE is a separate head of passing.

Assessment	Based on	Conducted by	Conduction and Marks Submission	Marks
LA1	Lab activities, attendance, journal	Lab Course Faculty	During Week 1 to Week 4 Submission at the end of Week 5	25
LA2	Lab activities, attendance, journal	Lab Course Faculty	During Week 5 to Week 8 Submission at the end of Week 9	25
LA3	Lab activities, attendance, journal	Lab Course Faculty	During Week 10 to Week 14 Submission at the end of Week 14	25
Lab ESE	Lab Performance and related documentation	Lab Course faculty	During Week 15 to Week 18 Submission at the end of Week 18	25

Week 1 indicates starting week of Semester.
Lab activities/Lab performance shall include performing experiments, mini-project, presentations, drawings, programming and other suitable activities, as per the nature and requirement of the lab course.
The experimental lab shall have typically 8-10 experiments.

Course Contents:

Assignments to be carried out in any RDBMS like ORACLE//DB2/SQL-Server/PostgreSQL:
Assignments include conceptual design using ER model, SQL and PL/SQL

1. Database Design using ER model
2. Database schema design
3. Database creation and applying integrity constraints
4. Study of DDL statements and data manipulation statements
5. Study of Basic SQL SELECT statement for displaying data from single table or multiple tables
6. Study of SQL constructs for aggregating data using group functions, sub-queries and complex queries
7. Study and Implementation of Triggers
8. Study and Implementation of Stored Procedures
9. Transaction isolation levels and Concurrency control
10. Few aspects of authorization such as creating and managing users, roles, granting and revoking of privileges
11. Implementation of B+ tree, hash index in C or C++

Title of the Course: Computer Network Lab **Course code: 5CS275**	L	T	P	Cr
	0	0	2	1

Desirable Requirements: Data Communication

Textbooks:

1. Richard Steven, "Unix network programming", for Socket Programming, Prentice Hall ,3rd edition, 2015
2. James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson Education,5th /6th edition, 2012/2013

References:

1. Jeffery S. Beasley, "Networking", New Riders Press, 2nd edition, 2008.
2. Larry L. Peterson, Bruce S. Davie "Computer Networks: A Systems Approach", The Morgan Kaufmann Series in Networking, 5th edition, 2011.

Course Objectives: The course is designed to give the practical view of various networking concepts and protocols using tools and simulators**.**

1. To dig up theoretical and practical knowledge in computer networks.
2. To distinguish and show how to design and analyze different types of communication protocols.
3. To interpret basic skills needed to write network application using socket interface.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Demonstrate** the practical aspect of networking related to the theoretical concepts.	3	Applying
CO2	Simulate, configure and **analyze** the network using networking tools.	4	Analyzing

CO-PO Mapping :

PO and PSO	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**-**
CO2	**-**	**-**	**-**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

1: Low, 2: Medium, 3: High

Assessments :

Lab Assessment:

There are four components of lab assessment, LA1, LA2, LA3 and Lab ESE.
IMP: Lab ESE is a separate head of passing.

Assessment	Based on	Conducted by	Conduction and Marks Submission	Marks
LA1	Lab activities, attendance, journal	Lab Course Faculty	During Week 1 to Week 4 Submission at the end of Week 5	25
LA2	Lab activities, attendance, journal	Lab Course Faculty	During Week 5 to Week 8 Submission at the end of Week 9	25
LA3	Lab activities, attendance, journal	Lab Course Faculty	During Week 10 to Week 14 Submission at the end of Week 14	25
Lab ESE	Lab Performance and related documentation	Lab Course faculty	During Week 15 to Week 18 Submission at the end of Week 18	25

Lab activities/Lab performance shall include performing experiments, mini-project, presentations, drawings, programming and other suitable activities, as per the nature and requirement of the lab course. The experimental lab shall have typically 8-10 experiments.

Course Contents:

At least 10 to 12 assignments should be conducted on following topics:

1. Study of Internetworking devices.
2. Study of basic networking commands and network configuration.
3. Study of packet capturing and analyzing tools on windows platform(e.g. Wireshark)
4. Wireshark Lab: Ethernet and ARP.
5. Wireshark Lab: 802.11
6. Configuration of network topology using packet tracer tool
7. Configuration of routing protocols
8. Configuration of IPv6 address using Packet Tracer
9. Capture and analyze TCP and UDP packet using Wireshark
10. Analyzing TCP connection and termination using Wireshark
11. Socket programming using TCP and UDP.
12. Wireshark Lab: HTTP, DNS

Title of the Course and Course code: Programming Lab 2 **Course Code: 5CS276**	L	T	P	Cr
	0	0	2	1

Desirable Requirements: Object Oriented Paradigm, Object Oriented Concept and basic implementation in C++.

Textbooks:

1. Cay S. Horstmann, Gary Cornell "Core Java Fundamentals Volume –I" (The Sun Microsystems Press Java Series), 10th Edition, March 2016
2. Cay S. Horstmann, Gary Cornell, "Core Java Volume – II" (The Sun Microsystems Press Java Series), 10th Edition, April 2017

References:

1. Herbert Schildt, "Java Complete Reference", McGraw Hill Education, 10th Edition, November 2017
2. Kathy Sierra and Bert Bates, "Oracle Certified Associates JAVA Standard Edition 8 Programmer I Exam Guide", McGraw Hill Education (Oracle Press), May 2017
3. Kathy Sierra and Bert Bates, "Oracle Certified Associates JAVA Standard Edition 8 Programmer II Exam Guide", McGraw Hill Education (Oracle Press), July 2018

Course Objectives :

Summary: Java is widely used in every corner of world and of human life. Java is not only used in software's but is also widely used in designing hardware controlling software components. JAVA programming language provides variety of data types, methods and some of them are included in syllabus. Learning Java serves as a good introduction to software development. Main objectives are as follows-

1. To inculcate the understanding of JAVA programming environment, basic object oriented programming with JAVA (JAVA version 1.8 and above or the latest java version)
2. To introduce selection of appropriate concepts of java programming such as static and non-static classes and access modifiers, user defined classes, collection, interface, exception handling, multi-threading, packages like – i/o, util, net, jdbc etc.
3. To infuse skills of integrating all components to build small java application for real world problem.

Course Learning Outcomes:

CO	**After the completion of the course the student should be able to**	Bloom's Cognitive	
		level	Descriptor
CO1	**Convert** the real world problem using simple java programing domain and identify the required java object oriented concept	2	Understanding
CO2	**Demonstrate** small application using java as a programing language for socio economic importance	3	Applying

CO-PO Mapping :

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

1: Low, 2: Medium, 3: High

Assessments :

Lab Assessment:

There are four components of lab assessment, LA1, LA2, LA3 and Lab ESE.
IMP: Lab ESE is a separate head of passing.

Assessment	Based on	Conducted by	Conduction and Marks Submission	Marks
LA1	Lab activities, attendance, journal	Lab Course Faculty	During Week 1 to Week 4 Submission at the end of Week 5	25
LA2	Lab activities, attendance, journal	Lab Course Faculty	During Week 5 to Week 8 Submission at the end of Week 9	25
LA3	Lab activities, attendance, journal	Lab Course Faculty	During Week 10 to Week 14 Submission at the end of Week 14	25
Lab ESE	Lab Performance and related documentation	Lab Course faculty	During Week 15 to Week 18 Submission at the end of Week 18	25

Week 1 indicates starting week of Semester.
Lab activities/Lab performance shall include performing experiments, mini-project, presentations, drawings, programming and other suitable activities, as per the nature and requirement of the lab course.
The experimental lab shall have typically 8-10 experiments.

Course Contents:	
Module 1 - An Introduction to Java	**3 Hrs.**
Features of JAVA language, Java Programming Environment-JDK, JRE, JVM, Fundamental Programming Structures in Java, Comparison of Java with C++, classes and objects, Inheritance and Interfaces. **Experiments:** **1.** Installation of jdk package, understand the difference between jdk and jre folder, set environment variable PATH/CLASSPATH. **2.** Simple hello word program for understanding – java structure, command and steps for executing java program. Also simple program for reading input from user using scanner class. **3.** Implementation of different inheritance types, Multiple Inheritance using Interface	
Module 2 - Fundamentals of Java	**5 Hrs.**
Use of keywords like final, static and abstract, Packages, String class, StringBuilder/StringBuffer class, Exception handling in Java **Experiments:** 1. Implementation of Package and access mechanism in package 2. String class implementation, basic operation, creating immutable and mutable string 3. Exception Handling	
Module 3 - I/O programming and util package	**6 Hrs.**
Hierarchy of classes in I/O Package, Streams: Character oriented and Byte oriented, Reading basic data types from keyboard, File handling in Java. Utility Methods for Arrays. The Collection Framework : List, Set, Map **Experiments:** 1. Implement collection utility classes – list, set, map with their specific methods available in interface or implemented class. 2. Implement exception related to IO and collection classes. 3. Program to read basic data types from keyboard using Scanner and check the entered values data type for its appropriateness	
Module 4 – Multithreading in Java	**3 Hrs.**
Multithreading – Classes support thread creation and execution Thread States & Synchronization of threads **Experiments:** 1. Multithreading – display thread information. 2. Multithreading – create thread using Thread and Runnable class. 3. Multithreading – thread communication and synchronization of threads.	
Module 5 - Java-Oracle database connectivity	**4 Hrs.**
Introduction to JDBC (Oracle Connectivity) JDBC Drivers & Initialization Creating Connection and Connecting to Databases CURD operation Using JDBC (Oracle Connectivity) Performing operations on specific row in oracle database using JDBC **Experiments:** 1. Design Database program for Employee details and implement INSERT, SELECT, DELETE, and UPDATE queries. 2. Implement ResultSet class. 3. Implement RowSet class.	

Module 6 - Graphical user interface in Java	**5 Hrs.**
GUI Design in Java – User Interface Event Handling in Java - Event delegation model (MVC model) Introduction to Swing. **Experiments:** 1. GUI design and Event handling 2. GUI design using Swing package - a) Celsius to Fahrenheit conversion b) Login and Password Verification. **3.** Implement exception related to event handling, GUI design.	

Presentation and Report Writing (5CS277)

EVEN Semester

EVEN Semester

Mandatory Life Skill Courses

Environmental Sciences (5IC201)

EVEN Semester

Minor Specialization Courses

Title of the Course: Software Engineering (Minor) Course Code: 1CSMO2	L	T	P	Cr
	3	0	0	3

Desired requirements:

Textbooks:

1. Pankaj Jalote, "An Integrated Approach to Software Engineering", Narosa Publishers, 3rd Edition, 2005.
2. Ian Sommerville, "Software Engineering", Addison-Wesley, 7th Edition, 2004.
3. James Rumbaugh, "Object Oriented Modeling and Design with UML", Pearson, 2nd Edition,2004.

References:

1. Roger S. Pressman, *"*Software Engineering: Practitioner's Approach*",* McGraw Hill, 7th Edition, 2010.
2. Jawadekar W.S., *"*Software Engineering: principles and practices*",* Tata McGraw Hills, 1st Edition.
3. Gillies A.C. and Smith p., *"*Managing Software Engineering: CASE studies and solutions*",* Chapman and Hall, London.

Course Objectives :

1. To unleash the orientation & importance of engineering approach to software development.
2. To infuse the knowledge of software processes & models practiced at IT industries.
3. To acquaint students with the SDLC phases in detail.
4. To emphasize on Design aspect with UML technology.
5. To inculcate the importance of software quality by virtue of software testing methods.

Course Learning Outcomes:

CO	After the completion of the course the student Learner be able to	Bloom's Cognitive level	Descriptor
CO1	Grasp industry processes on software development to become IT industry-savvy.	2	Understanding
CO2	Prepare with the spirit of team-working and importance of using artifacts at SDLC phases.	3	Applying
CO3	Distinguish and evaluate procedural & OO based development practices.	4	Analyzing
CO4	Integrate expertise on CASE tools usage especially for design and testing of software to undertake industrial strength software projects.	6	Creating

CO-PO Mapping :

PO and PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1			2								3	2	2	
CO2			1	2				2	2	3				
CO3					2									
CO4			2									2		3

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 pedagogy such as brainstorming, role play, quiz, presentations etc.
MSE: Assessment is based on 50% of course content (Normally first three modules)
ESE: Assessment is based on 100% course content with 70-80% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Module 1: Software Processes	**6 Hrs.**
Need of software engineering approach, ETVX model, Project management process, Software development process & models, Configuration management process, Process management process.	
Module 2: Software Quality & Project Planning	**2 Hrs.**
Quality objectives, PAF Model, Quality standards CMM & ISO, Project management plan, Cost estimation using COCOMO, Risk management.	
Module 3: Software Requirement Analysis & Function Oriented Design	**2Hrs.**
Software requirement process, Characteristics & Components of SRS. Design principles, Module level concepts.	
Module 4: Object Oriented Design with UML Diagrams	**2 Hrs.**

UML model, UML diagrams: Use-case, Class, Activity, State-chart, Interaction, Sequence, Collaboration, Component, Deployment.	
Module 5: Coding	**3Hrs.**
Best programming practices such as TDD & pair programming, verification.	
Module 6: Software Testing	**5Hrs.**
Testing purpose and concepts, test process, Levels of testing, Black Box and White Box Testing.	

Module wise Measurable Students Learning Outcomes :

The student should be able to:

Module 1: Software Processes

- Awareness of Software processes & Models used at IT.

Module 2: Software Quality & Project Planning

- Grasp quality parameters and standards such as PAF.
- Know & prepare project planning phases and responsibilities with WBS.

Module 3: Software Requirement Analysis & Function Oriented Design

- As per SDLC phase understand requirement process and need of SRS artifact. Understand functional & non-functional requirements as well. Realize the importance of design aspects, concepts & methodology. Practices to learn how to draw DFD on requirements.

Module 4:Object Oriented Design with UML & Continual Integration

- Building capability to draw & distinguish various UML diagrams on requirements. Articulating usage of Continual integration with Agile model process frameworks.

Module 5: User Interface Design & Coding

- Know the UI aspect of interactive design for enterprise applications.
- Learn best coding standards/practices such as TDD, pair programming and how to verify code.

Module 6: Software Testing

- Integrate expertise on how testing helps in quality of software. Know testing concepts, levels of testing. Learn and practice Black & white box testing along with test case generations using open- source tools.

This is Last Page