B. Tech. in Computer Engineering

Syllabus Aligned with New Education Policy 2020

With effective from 2025 - 2026

(University Department only)

Department of Computer Engineering

Dr. Babasaheb Ambedkar Technological University, Lonere - 402103

www.cse.dbatu.ac.in

B. Tech. in Computer Engineering and Computer Science & Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2023 - 2024)
First Semester

Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
		L	P	CA	MSE	ESE	
23UD1000BS101	Engineering Mathematics - I	4	-	20	20	60	4
23UD1CHEBS102	Engineering Chemistry	3	-	20	20	60	3
23UD1CHEBSL103	Engineering Chemistry Lab	-	2	60	-	40	1
23UD1245PC104	Set Theory and Logic	3	-	20	20	60	3
23UD1245VSL105	C Programming	1	2	60	-	40	2
23UD1245ES106	Digital Electronics	3	-	20	20	60	3
23UD1245ESL107	Digital Electronics Lab	-	2	60	-	40	1
23UD1245BS108	Environmental Science and Engineering	2	-	20	20	60	2
23UD1EEEESL109	Environmental Science and Engineering Lab	-	2	60	-	40	1
23UD1000CC112	A. NSS B. NCC C. Yoga Education	1	2	60	-	40	2
	Total	**17**	**10**	**340**	**100**	**460**	**22**

Course Type and Acronyms used

1. Basic Science Course (BSC)
2. Engineering Science Course (ESC)
3. Program Core Course (PCC)
4. Vocational Skill Enhancement Course (VSEC)
5. Co-curricular Course (CC)
6. Ability Enhancement Course (AEC)
7. Indian Knowledge System (IKS)

L - Lecture,
P / PR - Practical,
CA - Continuous Assessment, TH - Theory,
MSE - Mid Semester Examination,
ESE - End Semester Examination,
CR - Credit

B. Tech. in Computer Engineering and Computer Science & Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2023 - 2024)
Second Semester

Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
		L	P	CA	MSE	ESE	
23UD1000BS201	Engineering Mathematics- II	4	-	20	20	60	4
23UD1PHYBS202	Engineering Physics	3	-	20	20	60	3
23UD1PHYBSL203	Engineering Physics Lab	-	2	60	-	40	1
23UD1245BS204	Statistics and Probability	3	-	20	20	60	3
23UD1245PC205	OOP in C++	3	-	20	20	60	3
23UD1245PCL206	OOP Lab	-	2	60	-	40	1
23UD1245VEL207	Web Site Development and MS Office	1	2	60	-	40	2
23UD1000VS208	Communication Skills	2	-	20	20	60	2
23UD1000VSL209	Communication Skills Lab	-	2	60	-	40	1
23UD1245PCL210	Data Science with Python	-	2	60	-	40	1
23UD1245IK211	Hindustani Music	-	2	60	-	40	1
Total		**16**	**12**	**460**	**100**	**540**	**22**
Exit Requirements for Certificate Program (any two courses from following courses)							
23UD1245VE212	Mobile Application Development for iOS Devices	1	6	60	40	-	4
23UD1245VE213	Mobile Application Development for Android Devices	1	6	60	40	-	4
23UD1245VE214	Drone Technology	1	6	60	40	-	4
23UD1245CC212	Entrepreneurship	1	6	60	40		4

Course Type and Acronyms used

1. Basic Science Course (BSC)
2. Engineering Science Course (ESC)
3. Program Core Course (PCC)
4. Vocational Skill Enhancement Course (VSEC)
5. Co-curricular Course (CC)
6. Ability Enhancement Course (AEC)
7. Indian Knowledge System (IKS)
8. L - Lecture,
9. P / PR - Practical,
10. CA - Continuous Assessment, TH - Theory,
11. MSE - Mid Semester Examination,
12. ESE - End Semester Examination,
13. CR - Credit

B. Tech. in Computer Engineering and Computer Science & Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2024 - 2025)
Third Semester

Course Category	Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
			L	P	CA	MSE	ESE	
BSC	24UD1000BS301	Engineering Mathematics-III	3	-	20	20	60	3
PCC	24UD1245PC302	Data Structures and Algorithm	3	-	20	20	60	3
PCC	24UD1245PC303	Computer Architecture and Organization	3	-	20	20	60	3
ESC	24UD1245ES304	Data Communication	3	-	20	20	60	3
Open Elective	24UD1SCIOEM05A	Quantum Computing	2	-	20	20	60	2
MDM	24UD1245MD306	Python Programming	2	-	20	20	60	2
VSEC	24UD1UHVVE307	UHV - II	3	-	20	20	60	3
VSEC	24UD1245AE308A	Life of Chhatrapati Shivaji Maharaj	1	-	50	-	-	1
AEC	24UD1245AE309	Business Economics	2	-	20	20	60	2
PCC Lab	24UD1245PCL310	Data Structure and Algorithm Lab	-	2	40	-	60	1
PCC Lab	24UD1245PCL311	IDEA Lab	-	2	40	-	60	1
CEP/FP	24UD1245CP312	Seminar	-	2	40	-	60	1
		Total	**22**	**6**	**330**	**160**	**660**	**25**

Course Type and Acronyms used

14. Basic Science Course (BSC)
15. Engineering Science Course (ESC)
16. Program Core Course (PCC)
17. Vocational Skill Enhancement Course (VSEC)
18. Co-curricular Course (CC)
19. Ability Enhancement Course (AEC)
20. Indian Knowledge System (IKS)

L - Lecture,
P / PR - Practical,
CA - Continuous Assessment, TH - Theory,
MSE - Mid Semester Examination,
ESE - End Semester Examination,
CR - Credit

B. Tech. in Computer Engineering and Computer Science & Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2024 - 2025)
Fourth Semester

Course Categories	Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
			L	P	CA	MSE	ESE	
PCC1	24UD1245PC401	Design and Analysis of Algorithms	3	-	20	20	60	3
PCC2	24UD1245PC402	Discrete Mathematics	3	-	20	20	60	3
PCC3	24UD1245PC403	Database Management System	3	-	20	20	60	3
HSMM	24UD1245AE404	Principle of Management	2	-	20	20	60	2
OE	24UD1245OE405	Numerical Methods	2	-	20	20	60	2
MDM	24UD1245MD406	Data Structures	2	-	20	20	60	2
VSEC	24UD1COIVE407	Constitution of India	2	-	40	-	60	Audit
VSEC	24UD1000VE408B	Life of Bharat Ratna Dr. Babasaheb Ambedkar	1	-	50	-	-	1
VSEC	24UD1245S409	Competitive Programming	1	2	40	-	60	2
AEC	24UD1000AE410	Modern Indian Languages A) Marathi B) Hindi C) Sanskrit	2	-	20	20	60	2
VSEC	24UD1245S411	Full Stack Development	1	2	40	-	60	2
PCC Lab	24UD1245PCL412	Design and Analysis of Algorithms Lab	-	2	40	-	60	1
PCC Lab	24UD1245PCL413	Database Management System Lab	-	2	40	-	60	1
Total			**22**	**8**	**390**	**140**	**720**	**24**
	Exit Requirements for Certificate Program							
VSEC	23UD1245VE41	Full Stack Development Project	–	16	60	–	40	8

Course Type and Acronyms used

1. Basic Science Course (BSC)
2. Engineering Science Course (ESC)
3. Program Core Course (PCC)
4. Vocational Skill Enhancement Course (VSEC)
5. Co-curricular Course (CC)
6. Ability Enhancement Course (AEC)
21. Indian Knowledge System (IKS)

L - Lecture, TH - Theory,
P / PR - Practical,
CA - Continuous Assessment,
MSE - Mid Semester Examination,
ESE - End Semester Examination,
CR - Credit.

Third Year in B. Tech. in Computer Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2025 - 2026)
Fifth Semester

Course Type	Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
			L	P	CA	MSE	ESE	
PCC1	25UD1245PC501	Machine Learning	3	-	20	20	60	3
PCC2	25UD1245PC502	Theory of Computations	4	-	20	20	60	4
PCC3	25UD1245PC503	Operating System	3	-	20	20	60	3
PEC	25UD1245PE504	**Elective I:** A) Geographical Information Systems B) Embedded Systems C) Image Processing & Computer Vision	3	-	20	20	60	3
OE	25UD1245OE505	**Open Elective I:** A) Prompt Engineering B) Recommender System	3	-	20	20	60	3
MDM	25UD1245MD506	Introduction to Machine Learning	2	-	20	20	60	2
PCC Lab	25UD1245PL507	Machine Learning Laboratory	-	2	40	-	60	1
PCC Lab	25UD1245PL508	Operating System Laboratory	-	2	40	-	60	1
		Total	**18**	**4**	**200**	**120**	**480**	**20**

Third Year in B. Tech. in Computer Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2025 - 2026)
Sixth Semester

Course Type	Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
			L	P	CA	MSE	ESE	
PCC1	25UD1245PC601	Compiler Design	3	-	20	20	60	3
PCC2	25UD1245PC602	Computer Networks	3	-	20	20	60	3
PCC3	25UD1245PC603	Software Engineering	3	-	20	20	60	3
PEC	25UD1245PE604	Elective II: A. Internet of Things B. Advanced Algorithms C. Big Data Analytics	3	-	20	20	60	3
PEC	25UD1245OE605	Elective III: A. Distributed Systems B. Object Oriented Analysis Design C. Brain Computer Interface	3	-	20	20	60	3
MDM	25UD1245MD606	Data Analytics	2	-	20	20	60	2
PCC Lab	25UD1245PL607	Compiler Design Laboratory	-	2	40	-	60	1
PCC Lab	25UD1245PL608	Computer Networks Laboratory	-	2	40	-	60	1
PCC Lab	25UD1245PL609	Software Engineering Laboratory	-	2	40	-	60	1
VSEC	25UD1245VS610	Mini Project	-	4	40	-	60	2
Total			**17**	**10**	**280**	**120**	**600**	**22**
Exit Requirements for Certificate program								
VSEC	25UD1245VS611	A Data Analysis Project	–	16	60	–	40	8

Course Type and Acronyms used

1. Basic Science Course (BSC)
2. Engineering Science Course (ESC)
3. Program Core Course (PCC)
4. Vocational Skill Enhancement Course (VSEC)
5. Co-curricular Course (CC)
6. Ability Enhancement Course (AEC)
7. Indian Knowledge System (IKS)

L - Lecture,
P / PR - Practical,
CA - Continuous Assessment, TH - Theory,
MSE - Mid Semester Examination,
ESE - End Semester Examination,
CR - Credit

Fourth Year in B. Tech. in Computer Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2026 - 2027)
Seventh Semester

Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
		L	P	CA	MSE	ESE	
PCC1	Artificial Intelligence	3	-	20	20	60	3
PCC2	Cloud Computing	3	-	20	20	60	3
PEC	**Elective IV:** A. Natural Language Processing B. Research Methodology C. Software Testing	2	-	20	20	60	2
MDM	Deep Learning	2	-	20	20	60	2
Internship / OJT	Project Phase - I		24				12
	Total						22

Course Type and Acronyms used

1. Basic Science Course (BSC)
2. Engineering Science Course (ESC)
3. Program Core Course (PCC)
4. Vocational Skill Enhancement Course (VSEC)
5. Co-curricular Course (CC)
6. Ability Enhancement Course (AEC)
7. Indian Knowledge System (IKS)

L - Lecture,
P / PR - Practical,
CA - Continuous Assessment, TH - Theory,
MSE - Mid Semester Examination,
ESE - End Semester Examination,
CR - Credit

Fourth Year in B. Tech. in Computer Engineering
Course Curriculum Aligned with New Education Policy 2020
(with effective from Academic Year 2026 - 2027)
Eighth Semester

Course Code	Course Name	Weekly Hours		Examination Scheme			Credit
		L	P	CA	MSE	ESE	
PCC1	Software Architecture	3	-	20	20	60	3
PCC2	Cryptography and Network Security	3	-	20	20	60	3
PEC	**Elective V:** A. Blockchain Technology B. Virtual Reality C. Social Network Analysis	3	-	20	20	60	3
MDM	System Administration	2	-	40	-	60	2
Open Elective	Information Theory & Coding	3	2	20	20	60	4
Entrepreneurship / Economics / Management	Intellectual Property Rights	2	-	40	-	60	2
VSEC	Foreign Language Studies	2	-	40	-	60	2
AEC	Project Phase - II	4	-	40	-	60	2
	Total	**16**	**12**	**460**	**100**	**540**	**22**

Teaching Scheme		Semester V Machine Learning	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand core concepts and types of machine learning. 2. Apply supervised learning algorithms and handle overfitting. 3. Implement neural networks and basic deep learning models. 4. Explore probabilistic models and learning theory. 5. Use clustering and recommendation techniques for data analysis.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation, Linear regression, Decision trees, over fitting, Instance based learning, Feature reduction, Collaborative filtering based recommendation.	07 Hrs
2	Probability and Bayes learning, Logistic Regression, Support Vector Machine, Kernel function and Kernel SVM.	07 Hrs
3	Perceptron, multilayer network, back propagation, introduction to deep neural network.	07 Hrs
4	Computational learning theory, PAC learning model, Sample complexity, VC Dimension, Ensemble learning.	07 Hrs
5	Clustering k-means, adaptive hierarchical clustering, Gaussian mixture model.	07 Hrs

Text/Reference Books:

1. Tom Mitchell, Machine Learning, First Edition, McGraw Hill, 1997.
2. Ethem Alpaydin, Introduction to Machine Learning, Prentice Hall India Learning Private Limited; 2nd edition, 2010.

Teaching Scheme		Semester V Theory of Computations	Examination Scheme	
TH	3	**Course Objectives:** 1. Learn the fundamentals of finite automata and their applications. 2. Understand regular expressions and grammars with their relation to automata. 3. Analyze context-free grammars and their simplification techniques. 4. Study pushdown automata and context-free language recognition. 5. Explore Turing machines, undecidability, and the Chomsky hierarchy.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	**Finite Automata (FA):** Introduction, Deterministic Finite Automata (DFA) - Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.	08 Hrs
2	**Regular Expressions (RE):** Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions. **Regular Grammars:** Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular - Pumping lemma, applications, Closure properties of regular languages.	07 Hrs
3	**Context Free Grammar (CFG):** Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted).	07 Hrs
4	**Pushdown Automata:** Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. **Turing Machines (TM):** Formal definition and behaviour, Languages of a TM, TM as accepters and TM as a computer of integer functions, Types of TMs	07 Hrs
5	**Recursive And Recursively Enumerable Languages (REL):** Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language	07 Hrs

and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

Text/Reference Books:

1. Hopcroft, Ullman, Motwani, Introduction to Automata Theory, Languages, and Computation, Addison Wesley Publication, 2nd Edition, 2001.
2. Daniel I. A. Cohen, Introduction to Computer Theory, Wiley Publication, 1st Edition, 1986.
3. John C. Martin, Introduction to Languages and Theory of Computation, McGraw-Hill Publication, 4th Edition, 2010.
4. Krithivasan Kamala, Introduction to Formal Languages, Automata Theory and Computation, Pearson Education, 1st Edition, 2009.
5. Papadimitriou, Lewis, Elements of the Theory of Computations, PHI Publication, 2nd Edition, 1997.
6. E. V. Krishnamurthy, Introductory Theory of Computer Science, Springer-Velang New York Inc., 1st Edition, 1985.

Teaching Scheme		Semester V Operating Systems	Examination Scheme	
TH	3	**Course Objectives:**	CA	20
PR	-	1. Understand the structure, types, and components of operating systems, including system calls and virtual machines.	MSE	20
CR	3	2. Learn process management, CPU scheduling algorithms, and multithreading models.	ESE	60
		3. Analyze process synchronization, classical problems, and deadlock handling techniques.		
		4. Explore memory and virtual memory management, including paging and page replacement algorithms.		
		5. Study file systems, disk management, and mass-storage structures for efficient data handling.		

COURSE CONTENT

Unit No.	Topic	Hours
1	**Introduction and Operating system structures:** Definition, Types of Operating system, Real-Time operating system, System Components: System Services, Systems Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generations.	06 Hrs
2	**Processes and CPU Scheduling:** Process Concept, Process Scheduling, Operation on process, Inter-process Communication, Cooperating processes, Threads, Multithreading model, Scheduling criteria, Scheduling Algorithms, Thread Scheduling, Multiple-Processor Scheduling, Scheduling Algorithms evaluation.	06 Hrs
3	**Process Synchronization:** The critical-section problem, Critical regions, Peterson‘s Solution, Synchronization Hardware, Semaphores, Classical Problems of synchronization, and Monitors Deadlocks: Systems Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Combined approach to deadlock Handling.	06 Hrs
4	**Memory Management:** Basic concept, Logical and Physical address map, Memory allocation: Continuous Memory Allocation, Fixed and variable partition, Internal and external fragmentation and compaction, Paging: Principle of operation, Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging; Segmentation. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page / Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).	06 Hrs
5	**File Management:** File Concept, Access methods, File types, File operation, Directory and disk structure, File System Structure, File System Implementation, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table),	06 Hrs

efficiency and performance. Mass-Storage Structure: Disk Structure, Disk attachment, Disk scheduling, Disk management, Swap Space Management.

Text/Reference Books:

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, Operating System Concepts, Wiley Publication, 8th Edition, 2008.
2. Andrew S. Tanenbaum, Modern Operating System, PHI Publication, 4th Edition, 2015.
3. D. M. Dhamdhere, Systems Programming and Operating Systems, McGraw-Hill, 2nd Edition, 1996.
4. Garry Nutt, Operating Systems Concepts, Pearson Publication, 3rd Edition, 2003.
5. Harvey M. Deitel, An Introduction to Operating Systems, Addison Wesley Publication, 2nd Edition, 1990.
6. Thomas W. Doeppner, Operating System in Depth: Design and Programming, Wiley Publication, 2011.

Teaching Scheme		Semester V Geographical Information System	Examination Scheme	
TH	3	**Course Objectives:** 1. To understand the different components of GIS. 2. To understand the different raster data file formats. 3. To learn the Pre-processing of spatial datasets. 4. To understand various GIS analysis.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	What is Geographic Information Systems?, Different components of GIS, Different types of vector data, Raster data models and their types TIN data model.	06 Hrs
2	Advantages and disadvantages associated with vector, raster and TIN Non-spatial data attributes and their type Raster data compression techniques Different raster data file formats spatial database systems and their types.	06 Hrs
3	Pre-processing of spatial datasets Different map projections, Spatial interpolation techniques Different types of resolutions Digital Elevation Model (DEM).	06 Hrs
4	Quality assessment of freely available DEMS GIS analysis-1	06 Hrs
5	GIS analysis-2 and applications Errors in GIS Key elements of maps.	06 Hrs

Text/Reference Books:

1. Ian Heywood, Sarah Cornelius and Steve Carver, An Introduction to Geographical Information Systems (4th Edition) 2012.
2. Chang Kang-tsung (Karl), Introduction to Geographic Information Systems, 2006.
3. Tor Bernhardsen Geographic Information Systems: An Introduction, May 2002.

Teaching Scheme		Semester V Embedded Systems	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand the fundamentals of embedded systems, including design challenges, processor and IC technologies. 2. Learn the architecture and software of general-purpose and application-specific processors, with a focus on ARM7TDMI-S. 3. Explore system control features such as memory mapping, interrupts, timers, GPIO, and power management. 4. Study serial communication interfaces including UART, SPI, and I2C with their configurations and operations. 5. Apply RTOS concepts and design embedded systems through real-world examples and simulations using tools like Proteus VSM.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	**Introduction:** Embedded system overview, Design challenge, Processor technology, IC technology, Design technology, Custom single processor technology, Hardware-combinational logic, Sequential logic, Custom single purpose processor design, RT-level custom single purpose processor design, Optimizing custom single purpose processors.	06 Hrs
2	**General Purpose Processor Software:** Basic architecture, Operation, Programmers view, Development environment, Application specific instruction set processor, Selecting a microprocessor, General purpose processor design. Introduction, ARM7TDMI-S processor, Block diagram, Memory mapping, Memory accelerator module.	06 Hrs
3	**System Control:** Pin description, Register description, Crystal oscillator, External interrupt inputs, Other system controls, Memory mapping control, Phase locked loop, Power control, Reset, APB divider, Wakeup timer. GPIO: GPIO register map, Timer-TIMER / COUNTER0 and TIMER / COUNTER1 register map, Example timer operation, Architecture.	06 Hrs
4	**UART:** UART0/1 - UART0/1 register map, UART0/1 baud rate, UART0/1 auto-baud, UART0/1 block diagram. Serial peripheral interface: SPI data transfers, SPI pin description, SPI register map, SPI block diagram; I2C-bus interface: I2C bus configuration, I2C operating modes, I2C Bus serial interface block diagram, Summary of I2C registers.	06 Hrs
5	Introduction, Process scheduling, Examples of RTOS, Microprocessor and microcontroller based system design, typical design examples, system design and simulation using simulation software such as Proteus VSM. Digital Camera Example Introduction, Introduction to a Simple Digital Camera; User‘s Perspective, Designer‘s perspective requirements specification non-functional requirements, Informal functional specification, refined functional specification.	06 Hrs

Text/Reference Books:

1. Frank Vahid ―Embedded System Design- A Unified system Hardwar/Software Introduction‖, (3rd Edition, John Wiley India) ISBN 978-81-265-0837-2.
2. LPC 214x User manual (UM10139):- www.nxp.com
3. Andrew N. Sloss, Dominic Symes and Chris Wright ―ARM System Developer‘s Guide – Designing and Optimizing System Software‖, (Elsevier) ISBN: 1-55860-874-5.
4. LPC 17xx User manual (UM10360) :- www.nxp.com
5. ARM architecture reference manual : - www.arm.com
6. Steve Furber ―An Engineer‘s Introduction to the LPC2100 series‖ Trevor Martin (Hitex (UK) Ltd.).―ARM System-on-Chip Architecture‖ (2nd Edition, Addison-Wesley Professional)ISBN13: 9780201403527

Teaching Scheme		Semester V Image Processing and Computer Vision	Examination Scheme	
TH	3	**Course Objectives:**	CA	20
PR	-	1. To let the students learn the fundamental principles on the aspects of interdisciplinary research including acquiring, processing, analyzing, understanding and utilizing high-dimensional visual data from the real world.	MSE	20
CR	3	2. To equip the students with the knowledge of how to develop artificial intelligent systems which automate tasks that the human visual system can do.	ESE	60
		3. To guide the students to understand the relevant state of art technologies and gain experience throughout a variety of case studies		

COURSE CONTENT

Unit No.	Topic	Hours
1	**Introduction to Digital Image Processing:** Motivation & Perspective, Applications, Types of images, image file formats, Fundamentals Steps in Image Processing, Components of Image Processing System, Image digitization, Some basic relationships, Distance Measures between pixels, Image basic operation, Special Operations.	06 Hrs
2	**Image Enhancement and Transformation:** Image Enhancement: Introduction, Methods, Basic Intensity Transformation: Image Negatives, Log transformation, Power law Transformation, piecewise linear transformation functions, Histogram processing, Histogram Equalization and Matching. Basics of Spatial Filters, 2D Convolution & 2D Correlation, Smoothening (LPF) (Linear: Box, Gaussian & Non Linear: Median) and Sharpening (HPF): Laplacian operators, Unsharp Masking and Highboost Filtering, Combining Spatial Enhancement Methods. Image Transforms: Discrete Fourier transform (DFT): Definition and properties, FFT, DCT.	08 Hrs
3	**Morphological operations:** Introduction, erosion, dilation, opening, closing, Hit or Miss, boundary extraction, hole filling, connected components, the convex hull, thinning, thickening, skeletonization, and pruning	06 Hrs
4	**Segmentation and Feature Extraction:** Segmentation: Fundamentals; Point, Line and Edge Detection; Basics of edge detection: Image gradient and operators, Thresholding: Intensity Thresholding, Global thresholding, Segmentation by region growing, region splitting and merging. Feature Extraction: Boundary Preprocessing: Boundary Following (Tracing), Chain Codes (freeman & slope), Polygonal approximation, Signature, Boundary description: Shape number, Fourier Descriptor, Statistical Moments, Region Feature Descriptors: Topological feature, Texture.	08 Hrs
5	**Pattern Recognition:** Pattern and pattern classes, pattern classification by prototype matching (Minimum-Distance Classifier & using correlation for 2-D	07 Hrs

prototype matching), matching by structural prototype. Introduction to Bayes statistical classifiers, Introduction to Neural Network and Deep Learning.

Text Books:

1. Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall Robot Vision, by B. K. P. Horn, McGraw-Hill.
2. Introductory Techniques for 3D Computer Vision, by E. Trucco and A. Verri, Publisher: Prentice Hall.
3. R. C. Gonzalez, R. E. Woods. Digital Image Processing. Addison Wesley Longman, Inc., 1992.
4. Dhananjay K. Theckedath, Image Processing using MATLAB codes, Nandu Printers and Publishers Pvt. Ltd, Third edition.

Reference Books:

1. D. H. Ballard, C. M. Brown. Computer Vision. Prentice-Hall, Englewood Cliffs, 1982.
2. Richard Szeliski, Computer Vision: Algorithms and Applications (CVAA). Springer, 2010
3. Image Processing, Analysis, and Machine Vision. Sonka, Hlavac, and Boyle. Thomson.
4. Simon J. D. Prince, Computer Vision: Models, Learning, and Inference, Cambridge University Press, 2012
5. Mark Nixon and Alberto S. Aquado, Feature Extraction & Image Processing for Computer Vision, Third Edition, Academic Press, 2012.

Teaching Scheme		Semester V Prompt Engineering	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand the evolution of NLP and the fundamentals of large language models and their architecture. 2. Learn core prompt design techniques, including zero-shot, few-shot, and various prompt patterns. 3. Apply advanced prompting methods using APIs. 4. Design domain-specific prompts for tasks in coding, education, law, healthcare, and low-resource languages. 5. Address ethical considerations, safety, bias, and evaluation in prompt engineering and explore future trends.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	**Introduction to Prompt Engineering & LLMs:** Evolution of NLP to Large Language Models, Anatomy of LLMs: Tokens, embeddings, context windows, Why prompt engineering matters, Prompt engineering applications in AI/ML systems, Introduction to GPT, Claude, PaLM, Mistral, etc., Prompting vs fine-tuning.	06 Hrs
2	**Foundations of Prompt Design:** Types of prompting: Zero-shot, One-shot, Few-shot, Prompt patterns: Instructional, Delimiting, Role-based, Socratic, Prompt evaluation metrics: accuracy, relevance, hallucination, safety, Prompt debugging techniques	06 Hrs
3	**Advanced Prompting Techniques:** Chain-of-Thought Prompting, Self-Ask Prompting, ReAct (Reason + Act) prompting, Multimodal prompting (text + image), Prompt templates using Python APIs (LangChain, OpenAI SDK).	06 Hrs
4	**Prompt Engineering for Domain-Specific Tasks:** Coding assistance and code generation prompts, Conversational agents and dialogue design, Legal, educational, medical prompts, Data wrangling and analysis via prompting, Prompting in low-resource languages	06 Hrs
5	**Prompt Safety, Bias, and Ethics:** Bias, toxicity, misinformation in model outputs, Adversarial prompting and jailbreak attempts, Safety best practices in prompt design, Evaluation frameworks (BLEU, ROUGE, BERTScore, human evals), Future directions: function calling, tool use, retrieval-augmented generation (RAG)	06 Hrs

Text/Reference Books:

1. **"The Art of Prompt Engineering with OpenAI APIs"** by Nathan Hunter.
2. **"Prompt Engineering for Everyone"** by Isa Fulford & Andrew Ng (DeepLearning.AI short course).
3. **"Building AI Applications with OpenAI API"** by Yash Sheth.

Teaching Scheme		Semester V Recommender System	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand basics concepts of Recommended System. 2. Apply various types of recommendation system. 3. Evaluate recommendation system.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	Introduction, Goals of Recommender Systems, Basic Models / types of Recommender Systems, Challenges in Recommender Systems, The Cold-Start Problem in Recommender Systems, Attack-Resistant Recommender Systems, Privacy in Recommender Systems.	06 Hrs
2	**Collaborative Filtering:** Types of Collaborative Filtering, Neighbourhood / memory based vs Model based. Neighbourhood based Collaborative Filtering: User based Collaborative Filtering, Item based Collaborative Filtering, cold-start problem. **Model based Collaborative Filtering:** Naive Bayes Collaborative Filtering, Matrix Factorization, Singular Value Decomposition, Association rule mining.	06 Hrs
3	**Content-Based Recommender Systems:** Introduction, Basic Components of Content-Based Systems, Preprocessing and Feature Extraction, Learning User Profiles and Filtering, Content-Based Versus Collaborative Recommendations, High level architecture of content-based systems, Advantages and drawbacks of content based filtering, Item profiles, Discovering features of documents, Obtaining item features from tags, Representing item profiles, Methods for learning user profiles, Similarity measures, Similarity based retrieval, Classification algorithms. **Knowledge based recommendation:** Knowledge representation and reasoning, Constraint based recommenders, Case based recommenders	06 Hrs
4	**Time- and Location-Sensitive Recommender Systems:** Introduction, Temporal Collaborative Filtering, Discrete Temporal Models, Location - Aware Recommender Systems.	06 Hrs
5	**Evaluating Recommender Systems:** Introduction, Evaluation Paradigms, General Goals of Evaluation Design, Design Issues in Offline Recommender Evaluation, Accuracy Metrics in Offline Evaluation, Limitations of Evaluation Measures.	06 Hrs

Text/Reference Books:

1. Jannach D., Zanker M. and FelFering A., Recommender Systems: An Introduction, Cambridge University Press(2011), 1st ed.
2. Aggarwal, C. C. “Recommender Systems: The Textbook”. Springer 2016. ISBN 978-3-319-29657-9.
3. Deepak K. Agarwal, Bee-Chung Chen, ,Statistical Methods for Recommender Systems, Cambridge University Press (2016).

Teaching Scheme		Semester V Machine Learning Laboratory	Examination Scheme	
TH	-		CA	40
PR	-		MSE	-
CR	1		ESE	60

List of Experiments:

1. Implement a toy classification problem using decision boundaries (manual coding). Apply train/test split and k-fold cross-validation to a dataset (e.g., Iris). Compare accuracy, precision, recall, F1-score.
2. Implement simple and multiple linear regression using NumPy and using scikit-learn. Plot regression lines and residuals. Compute RMSE and R^2 values.
3. Train a decision tree classifier and visualize the tree. Analyze performance with/without pruning and depth control. Demonstrate overfitting using synthetic data.
4. Implement and visualize k-Nearest Neighbors classification. Tune k and compare accuracy. Demonstrate curse of dimensionality.
5. Apply PCA and t-SNE for dimensionality reduction. Visualize high-dimensional data in 2D. Measure accuracy before and after feature reduction.
6. Implement a simple user-based and item-based collaborative filter. Use cosine similarity or Pearson correlation. Recommend top-N items for a user.
7. Implement Gaussian Naive Bayes manually and using scikit-learn. Classify spam/ham email or sentiment analysis.
8. Train logistic regression for binary classification. Visualize the sigmoid function and decision boundary. Analyze ROC curve and AUC.
9. Train a linear and kernel (RBF, polynomial) SVM. Visualize margins and support vectors. Tune C and gamma hyperparameters.
10. Implement a single-layer perceptron from scratch. Train a multilayer perceptron using scikit-learn / Keras. Compare performance with different activation functions.
11. Visualize layers, activations, and loss using Keras. Build a 2–3 layer DNN for digit classification.
12. Simulate sample complexity: How performance improves with increasing training size. Discuss PAC model using toy examples. Conceptual lab: VC Dimension estimation for linear classifiers.
13. Compare Bagging (Random Forest) vs Boosting (AdaBoost/XGBoost). Visualize feature importance.
14. Apply K-Means and visualize clusters. Use silhouette score and elbow method to tune K.
15. Fit a Gaussian Mixture Model (GMM) and visualize decision boundaries. Perform Agglomerative Clustering and plot dendrograms.

Teaching Scheme		Semester V Operating System Laboratory	Examination Scheme	
TH	-		CA	40
PR	2		MSE	–
CR	1		ESE	60

List of Experiments:

1. Hands on Unix Commands
2. Shell programming for file handling.
3. Shell Script programming using the commands grep, awk, and sed.
4. Implementation of various CPU scheduling algorithms (FCFS, SJF, Priority).
5. Implementation of various page replacement algorithms (FIFO, Optimal, LRU).
6. Concurrent programming; use of threads and processes, system calls (fork and v-fork).
7. Study pthreads and implement the following: Write a program which shows the performance.
8. Improvement in using threads as compared with process.(Examples like Matrix Multiplication.
9. Hyper Quick Sort, Merge sort, Traveling Sales Person problem).
10. Implementation of Synchronization primitives – Semaphore, Locks and Conditional Variables.
11. Implementation of Producer-Consumer problem, Bankers algorithm.
12. Implementation of various memory allocation algorithms, (First fit, Best fit and Worst fit), Disk.
13. Scheduling algorithms (FCFS, SCAN, SSTF, C-SCAN).
14. Kernel reconfiguration, device drivers and systems administration of different operating systems.
15. Writing utilities and OS performance tuning.

Semester - VI

Teaching Scheme		Semester VI Compiler Design	Examination Scheme	
TH	3	**Course Objectives:**	CA	20
PR	-	1. Understand the phases of a compiler and tools used in compiler construction.	MSE	20
CR	3	2. Learn lexical analysis techniques and design of lexical analyzers.	ESE	60
		3. Explore syntax analysis methods including top-down, bottom-up, and LR parsing.		
		4. Apply syntax-directed translation for generating intermediate code and handling expressions.		
		5. Study code generation techniques, optimization, and runtime storage management.		

COURSE CONTENT

Unit No.	Topic	Hours
1	**Introduction to Compiling:** Definition, analysis of the source program, the phases of a compiler, the grouping of phases, Compiler Construction tools, A simple one-pass compiler.	06 Hrs
2	**Lexical Analysis:** The role of the Lexical analyzer, Input buffering, Specification of Tokens, A Language for Specifying Lexical Analyzers, Design of a Lexical Analyzer generator.	06 Hrs
3	**Syntax Analysis:** The role of the Parser, Context-free grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Operator-precedence Parsing, LR Parsers, Using Ambiguous Grammars, Parser Generators.	06 Hrs
4	**Syntax-Directed Translation:** Definitions, Construction of Syntax Trees, Bottom-Up Evaluation of S- Attributed definitions, Top-Down Translation, Bottom-Up Evaluation of Inherited attributes, Intermediate Languages, Declarations, Assignment Statements, Boolean Expressions, Case Statements, Back patching, Procedure Calls.	06 Hrs
5	**Code Generation:** Issues in the Design of a Code Generator, The target Machine, Run-Time Storage Management, Basic Blocks and Flow Graphs, Next-Use Information, Simple Code Generator, Register allocation and Assignment, The DAG Representation of Basic Blocks, Generating Code from DAGs, Dynamic Programming, Code-Generation Algorithm, Code-Generators.	06 Hrs

Text/Reference Books:

1. Aho, Sethi, Ullman, Compilers Principles, Techniques and Tools, Pearson Education India, 2nd Edition, 2013.
2. Hopcroft, Motwani and Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Publication, 2nd Edition, 2001.
3. Dick Grune, Kees van Reeuwijk, Henri E. Bal, Ceriel J. H. Jacobs and Koen Langendoen, Modern Compiler Design, Springer, 2nd Edition, 2012.

Teaching Scheme		Semester VI Computer Networks	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand the fundamentals of computer networks, reference models, and performance metrics. 2. Learn various LAN and wireless technologies including Ethernet, Wi-Fi, Bluetooth, and Wi-MAX. 3. Explore data link layer design, error detection/correction, and flow control mechanisms. 4. Study network layer protocols, routing algorithms, congestion control, and QoS techniques. 5. Gain knowledge of application layer protocols and basic network security principles.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	Applications of computer networks, Network hardware, Network software: Protocol Hierarchy, Design Issue, connection oriented vs. connectionless, Service Primitives, Reference models: OSI and TCP/IP, Example networks: Internet, Network standardization, Performance: Bandwidth and Latency, Delay and bandwidth product, High- Speed Network, Application Performance Needs.	06 Hrs
2	**LAN Technologies:** X5, Frame relay, ATM, Ethernet (802.3), FDDI, Token Rings, Resilient Packet Rings, Wireless LANs: Wi-Fi (802.11), Cell Phone Technologies, Broadband Wireless: Wi-MAX (802.16), Bluetooth (802.15.1), RFID.	06 Hrs
3	**Data Link Layer:** Data Link Layer Design Issues: Service provided to network layer Framing, Error Control, Flow Control, Error Detection and Correction: error correcting codes, error detecting codes.	06 Hrs
4	**Network Layer and Congestion Control:** IPv4/IPv6, Routers and Routing Algorithms distance vector link state. TCP UDP and sockets, General principles, Congestion prevention policies, Load shading, Jitter control, Quality of service: Packet scheduling, Traffic shaping, integrated Services.	06 Hrs
5	**Application Layer Protocols:** DNS, SMTP, POP, FTP, HTTP. Network Security: Authentication, Basics of public key and private key cryptography, digital signatures and certificates, firewalls	06 Hrs

Text/Reference Books:

1. A. Tanenbaum, Computer Networks, PHI Publication, 5th Edition, 2011.
2. B. Forouzan, Data Communications and Networking, McGraw Hill Publication, 5th Edition, 2013.
3. Larry Peterson and Bruce Davie, Computer Networks: A Systems Approach, Morgan Kufman Publication, 5th Edition, 2012.
4. D. Comer, Computer Networks and Internet, Pearson Education, 6th Edition, 2014.

5. M. Gallo, W. Hancock, Computer Communications and Networking Technologies, Brooks/Cole Publisher, 2001.
6. Natalia Olifer, Victor Olifer, Computer Networks: Principles, Technologies and Protocols for Network Design, Wiley Publication, 2005.

Teaching Scheme		Semester VI Software Engineering	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand professional software development practices and ethics through case studies. 2. Learn software process models, including agile and plan-driven approaches. 3. Master requirements engineering and system modeling techniques. 4. Apply architectural design principles and object-oriented design using UML and design patterns. 5. Explore software testing methods, dependability, and security considerations.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	Professional software development, Software engineering ethics, Case studies. Software processes: Software process models, Process activities, Coping with change, The rational unified process.	06 Hrs
2	**Agile software development:** Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods. Requirements engineering: Functional and non-functional requirements, The software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management.	06 Hrs
3	**System modeling:** Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering. Architectural design: Architectural design decisions, Architectural views, Architectural patterns, Application architectures.	06 Hrs
4	Design and implementation, Object-oriented design using UML, Design patterns Implementation issues, Open source development.	06 Hrs
5	Software testing, Development testing, Test-driven development, Release testing, User testing. Dependability properties, Availability and reliability, Safety Security.	06 Hrs

Text/Reference Books:

1. Ian Sommerville, Software Engineering; 9th Edition, Addison-Wesley Publishing Company, USA.
2. S.A. Kelkar, Software Engineering, , Prentice Hall of India, 2007.
3. Pressman, Software Engineering, Tata McGraw Hill, 6th Edition, 2006.
4. Pankaj Jalote, Software Engineering, Narosa Publishers, 3rd Edition, 2006.

Teaching Scheme		Semester VI Internet of Things	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand the evolution, architecture, and challenges of IoT and its impact on digitization. 2. Learn about smart objects, sensors, actuators, and IoT access technologies. 3. Explore IP networking, transport layers, and application protocols optimized for IoT. 4. Apply data analytics, machine learning, and security principles in IoT environments. 5. Gain hands-on experience with IoT devices like Arduino and Raspberry Pi and study smart city applications.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	**IoT Introduction:** Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT Functional Stack, IoT Data Management and Compute Stack.	06 Hrs
2	**Smart Objects:** The ―Things‖ in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access Technologies.	06 Hrs
3	**IP Layer:** IP as the IoT Network Layer, The Business Case for IP, The need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT, The Transport Layer, IoT Application Transport Methods.	06 Hrs
4	**Data and Analytics for IoT:** An Introduction to Data Analytics for IoT, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, Securing IoT, A Brief History of IoT Security, Common Challenges in IoT Security, Formal Risk Analysis Structures: OCTAVE and FAIR, The Phased Application of Security in an Operational Environment	06 Hrs
5	**IoT Physical Devices and Endpoints:** Building iot with Arduino: Arduino–Interfaces-Arduino IDE–Programming, RaspberryPi: Introduction to RaspberryPi, About the RaspberryPi Board: Hardware Layout, Operating Systems on RaspberryPi, Configuring RaspberryPi, Programming RaspberryPi with Python, Wireless Temperature Monitoring System Temperature Sensor, Connecting Raspberry Pi via SSH, Accessing Temperature from sensors, Remote access to RaspberryPi, Smart and Connected Cities, An IoT Strategy for Smarter Cities, Smart City IoT Architecture, Smart City Security Architecture, Smart City Use-Case Examples.	06 Hrs

Text/Reference Books:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet Things‖, 1st Edition, Pearson Education.
2. Srinivasa K G, ―Internet of Things‖, CENGAGE Leaning India, 2017.
3. Vijay Madisetti and Arshdeep Bahga, ―Internet of Things (A Hands-on-Approach), 1st Edition, VPT, 2014.
4. Raj Kamal, ―Internet of Things: Architecture and Design Principles‖, 1st Edition, McGraw Hill Education, 2017.

Teaching Scheme		Semester VI Advanced Algorithm	Examination Scheme	
TH	3	**Course Objectives:**	CA	20
PR	-	1. Master advanced graph algorithms including strongly connected components, network flows, and bipartite matching.	MSE	20
CR	3	2. Understand approximation and randomized algorithms with performance guarantees and key examples.	ESE	60
		3. Explore parameterized complexity, fixed-parameter tractability, kernelization, and exact algorithms for NP-hard problems.		
		4. Apply advanced dynamic programming techniques and optimizations on trees, graphs, and state spaces.		
		5. Learn fundamentals of parallel, distributed, and online algorithms with practical models and applications.		

COURSE CONTENT

Unit No.	Topic	Hours
1	**Advanced Graph Algorithms:** Strongly Connected Components (Kosaraju's Algorithm, Tarjan's Algorithm); Network Flow Algorithms: Ford-Fulkerson, Edmonds-Karp; Bipartite Graph Matching (Hungarian Algorithm); Topological Sorting Variants and Applications.	06 Hrs
2	**Approximation and Randomized Algorithms:** Concept of Approximation Algorithms and Performance Guarantees; Approximation for Vertex Cover, Set Cover, and TSP; Probabilistic Algorithms (Monte Carlo, Las Vegas); Randomized QuickSort and Skip Lists; Randomized Min-Cut Algorithm (Karger's Algorithm); Hashing with Randomization (Universal Hashing).	06 Hrs
3	**Parameterized and Exact Algorithms:** Introduction to Parameterized Complexity; Fixed-Parameter Tractability (FPT); Kernelization Techniques; Exact Algorithms for NP-Hard Problems (Branch and Bound refinements) Case Studies: Vertex Cover, Feedback Vertex Set, Solving SAT with DPLL Algorithm and Heuristics.	06 Hrs
4	**Advanced Dynamic Programming and Memoization:** DP on Trees and Graphs (Tree DP, DAG DP); Bitmask Dynamic Programming; State Space Reduction Techniques; Sliding Window Optimization, Divide and Conquer DP; Space-Optimized DP, Knuth-Yao Optimization; Convex Hull Trick and Monotonic Queue Optimization.	06 Hrs
5	Parallel, Distributed and Online Algorithms: Introduction to Parallel Algorithms (PRAM Models); Prefix Sum and List Ranking; Parallel Sorting and Matrix Multiplication; MapReduce Model (High-Level View); Introduction to Online Algorithms: Competitive Analysis; Paging, Caching, Ski Rental Problem.	06 Hrs

Text/Reference Books:

1. Algorithm Design, by Jon Kleinberg and Éva Tardos
2. Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein (CLRS)

3. Randomized Algorithms, by Rajeev Motwani and Prabhakar Raghavan
4. Approximation Algorithms, by Vijay Vazirani
5. Parameterized Algorithms, by Cygan et al.

Teaching Scheme		Semester VI Big Data Analytics	Examination Scheme	
TH	3	**Course Objectives:**	CA	20
PR	-	1. Understand the fundamentals, challenges, and applications of Big Data and its enabling technologies.	MSE	20
CR	3	2. Learn Big Data platforms like Apache Spark, HDFS, MapReduce, and distributed consistency models.	ESE	60
		3. Explore real-time data processing using streaming platforms such as Spark Streaming and Kafka.		
		4. Apply Big Data machine learning techniques with tools like Mahout and Spark MLlib.		
		5. Gain hands-on knowledge of modern NoSQL databases with MongoDB and query design.		

COURSE CONTENT

Unit No.	Topic	Hours
1	**Introduction to Big Data:** Why Big Data and Where did it come from?, Characteristics of Big, Challenges and applications of Big Data, Enabling Technologies for Big Data, Big Data Stack, Big Data distribution packages.	06 Hrs
2	**Big Data Platforms:** Overview of Apache Spark, HDFS, YARN, MapReduce, MapReduce Programming Model with Spark, MapReduce Example: Word Count, Page Rank etc, CAP Theorem, Eventual Consistency, Consistency Trade-O-s, ACID and BASE, Zookeeper and Paxos, Cassandra, Cassandra Internals, HBase, HBase Internals.	06 Hrs
3	**Big Data Streaming Platforms:** Big Data Streaming Platforms for Fast Data, Streaming Systems, Big Data Pipelines for Real-Time computing, Spark Streaming, Kafka, Streaming Ecosystem.	06 Hrs
4	**Big Data Applications:** Overview of Big Data Machine Learning, Mahout, Big Data Machine learning Algorithms in Mahout-kmeans, Naive Bayes etc. Machine learning with Spark, Machine Learning Algorithms in Spark, Spark MLlib, Deep Learning for Big Data, Graph Processing: Pregel, Giraph, Spark GraphX.	06 Hrs
5	**Database for the Modern Web:** Introduction to mongoDB key features, Core server tools, MongoDB through the JavaScript‘ sshell, Creating and querying through Indexes, Document-oriented, principles of schema design, Constructing queries on databases, collections and documents, MongoDB query language.	06 Hrs

Text/Reference Books:

1. Bart Baesens ―Analytics in a Big Data World: The Essential Guide to Data Science and its Applications‖, Wiley and SAS Business Series.
2. Rajkumar Buyya, Rodrigo N. Calheiros, Amir M Vahid Dastjerdi, Morgan Kaufmann, ―Big Data Principals and Paradiagram‖, Elsevier, ISBN: 978-0-12-805394-2.
3. 2Kyle Banker, Peter Bakkum and Shaun Verch, ―MongoDB in Action,

2nd Edition Dream tech Press, ISBN: 978-9351199359.

4. Anand Rajaraman, Jeffrey D. Ullman, ―Mining of Massive Datasets, 3rd edition,Cambridge University Press
5. Sima Acharya, Subhashini Chhellappan, ―BIG Data and Analytics, ,Willey publication, ISBN: 978-8126554782.

Teaching Scheme		Semester VI Distributed Systems	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand the evolution, models, and features of distributed computing systems and message-passing. 2. Learn Remote Procedure Call (RPC) concepts, implementation, and case studies. 3. Explore distributed shared memory architectures, consistency models, and synchronization techniques. 4. Study resource and process management including scheduling, load balancing, and migration. 5. Analyze distributed file system design, access models, fault tolerance, and real-world case studies.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	Introduction to Distributed Computing System, Evolution of Distributed Computing System, Distributed Computing System models, Distributed Computing System Gaining Popularity, Distributed Operating System, Introduction to Distributed Computing Environment (DCE), Desirable Features of a Good Message-Passing System, Issues in IPC by Message-Passing, Synchronization, Buffering, Multidatagram message, Encoding and Decoding of message data, Process addressing, Failure Handling, Group Communication, Case Study: BSD UNIX IPC Mechanism.	07 Hrs
2	**Remote Procedure Calls:** RPC model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC messages, Marshaling arguments and Results, Server Management, Parameter Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client- Server Binding, Exception Handling, Security, Some Special Types of RPCs, Case studies: Sun RPC, DCE, RPC.	06 Hrs
3	**Distributed Shared Memory:** General Architecture of DSM Systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other Approaches to DSM, Heterogeneous DSM, Advantages of Synchronization: Clock Synchronization, Event Ordering, Mutual Exclusion, Deadlock, Election Algorithms.	06 Hrs
4	**Resource Management And Process Management:** Desirable Features of a Good Global Scheduling Algorithm, Task assignment Approach, Load-Balancing Approach, load Sharing Approach, Process Migration, Threads.	06 Hrs
5	**Distributed File System:** Desirable Features of a Good Distributed File System, File Models, File Accessing Models, File Sharing Semantics, File Caching Schemes, File Replication, Fault Tolerance, Atomic Transactions, Design Principles, Case Study: DCE Distributed File Service.	06 Hrs

Text/Reference Books:

1. P. K. Sinha, Distributed Operating System, PHI Publication
2. Colorouis, Distributed Systems, Addison Wesley Publication.
3. M. L. Liu, Distributed Computing: Principles and Applications, Addison-Wesley, 2004.

Teaching Scheme		Semester VI Object Oriented Analysis Design	Examination Scheme	
TH	3	**Course Objectives:** 1. Understand core object-oriented concepts, principles, and system development life cycle. 2. Learn UML modeling techniques including class, sequence, use case, and activity diagrams. 3. Apply use case modeling and CRC cards for requirements gathering and design refinement. 4. Explore object-oriented design principles, class relationships, and refactoring practices. 5. Study design patterns, their application, and transition from design to implementation in OOP.	CA	20
PR	-		MSE	20
CR	3		ESE	60

COURSE CONTENT

Unit No.	Topic	Hours
1	Introduction Overview of object oriented system, Object orientation, Objects, attributes, object behavior, Object respond to messages, encapsulation, Inheritance, Polymorphism, object relationships and association, aggregation, Object identity static and dynamic binding, Object persistence, meta classes. Object oriented system development life cycle.	06 Hrs
2	Object oriented modeling Modeling, UML Modeling, class diagram, activity diagram, Sequence diagram, collaboration diagram state chart diagram, interaction diagram, Implementation diagram, use case diagram.	06 Hrs
3	Introduction to Use Cases, Use cases vs traditional functional requirements, Elements of Use Case Modeling, Use Case Diagrams (UML), Use Case Documentation, Use Case Realization, Use Case-Based Requirements Gathering, Introduction to CRC Cards, Structure of a CRC card, Identifying Classes from Use Cases, Defining Responsibilities, Identifying Collaborators, Creating and Using CRC Cards, CRC in Design Refinement, Case Study: Real-world example: Library system / Online shopping / Ticket booking.	06 Hrs
4	Introduction to Object-Oriented Design, Object-Oriented Design Principles, Design Process and Modeling, Identifying Classes and Relationships, Object Interaction and Communication, Refactoring and Code Smells. Introduction to Design Patterns, Classification of Patterns, Applying Design Patterns, Anti-Patterns and Pitfalls, Design Patterns in Modern Development: Patterns in frameworks (Spring, .NET, etc.), Use in Agile and Test-Driven Development (TDD)	06 Hrs
5	Implementation from Design to Implementation, Programming Style, Object-Oriented languages, Non-Object-Oriented languages, Object Oriented Databases, Computer animation, Electrical Distribution design System, Future of Object-Oriented Technology.	06 Hrs

Text/Reference Books:

1. Grady, Booch, Object Oriented analysis and design with applications, 2nd Edition, PHI.
2. James Rumbaugh, Object-Oriented Modeling And Design, 1st Edition, PHI Publication.
3. Ali Bahrami, Object Oriented Systems Development, 1st Edition, McGraw-Hill Publication.
4. Robert Lafore, Object Oriented Programming, Galgotia Publication.
5. Dan Pilone, Neil Pitman, UML 2.0 in a Nutshell: A Desktop Quick Reference, O'Reilly Media.
6. E. Balagurusamy, Object Oriented Programming, McGraw-Hill Publication.
7. S. Koshafian, Object Orientation, Wiley Publication.

Teaching Scheme		Semester VI Brain Computer Interface	Examination Scheme	
TH	3	**Course Objectives:**	CA	20
PR	-	1. Understand BCI fundamentals, types, brain structure, and ethical considerations.	MSE	20
CR	3	2. Learn bio-signal acquisition methods, hardware, and preprocessing techniques.	ESE	60
		3. Explore feature extraction, selection, and machine learning algorithms for BCI.		
		4. Study diverse BCI applications including neuroprosthetics, communication, and gaming.		
		5. Analyze emerging trends, challenges, and security issues in BCI technology.		

COURSE CONTENT

Unit No.	Topic	Hours
1	Introduction to BCI: Definition, purpose, and applications; History and evolution of BCIs; Types of BCIs: Invasive, non-invasive, partially invasive; Human brain structure and function: Neurons, synapses, and brain waves; EEG: Basics, signal types, and electrode placements (10-20 system); Ethical, legal, and societal implications of BCI.	06 Hrs
2	Bio-signal acquisition: EEG, ECoG, MEG, fNIRS; Noise sources: EMG, EOG, motion artifacts; Signal preprocessing techniques: Filtering (Band-pass, Notch), Artifact removal (ICA, PCA), Signal normalization and segmentation; Hardware for BCI: EEG headsets and amplifiers.	06 Hrs
3	Feature extraction techniques: Time-domain, frequency-domain, and time-frequency analysis, Wavelet transform, Fourier transform, Power spectral density; Feature selection and dimensionality reduction (PCA, LDA); Machine learning in BCI: Supervised and unsupervised methods, Common classifiers: SVM, k-NN, Decision Trees, Neural Networks; Performance metrics: Accuracy, Precision, Recall, F1-score, ROC.	06 Hrs
4	Applications of BCI: Neuroprosthetics and assistive devices, Communication aids for ALS patients, Gaming, Virtual Reality, Smart Home integration, Neurofeedback and cognitive training; Brain-to-brain interface (BBI) and collaborative BCI; Feedback mechanisms: Visual, auditory, haptic; Real-time systems and closed-loop control.	06 Hrs
5	Emerging trends: Deep learning in BCI, Hybrid BCI systems (EEG + fNIRS), Wireless and wearable BCI devices; Challenges in BCI: Signal variability, Inter-subject differences, Real-time processing and latency; Security and privacy in BCI.	06 Hrs

Text/Reference Books:

1. Jonathan Wolpaw & Elizabeth Winter Wolpaw, Brain-Computer Interfaces: Principles and Practice, Oxford University Press

2. Rajesh P. N. Rao, Brain-Computer Interfacing: An Introduction, Cambridge University Press

Teaching Scheme		Semester VI Compiler Design Laboratory	Examination Scheme	
TH	-		CA	40
PR	2		MSE	–
CR	1		ESE	60

List of Experiments:

1. Implement a one-pass compiler for a subset of a language. Write a simple program that reads arithmetic expressions and prints each token (manually).
2. Install and configure LEX and YACC tools. Demonstrate a basic program that uses LEX for token identification.
3. Use LEX/Flex to implement a lexical analyzer that identifies tokens: keywords, identifiers, numbers, operators, delimiters. Print each token along with its type.
4. Modify the lexical analyzer to eliminate comments and unnecessary whitespaces from input source code.
5. Write grammar rules for arithmetic expressions (support +, –, *, /). Implement top-down parsing using YACC/Bison with LEX as the lexical analyzer.
6. Write a C/Python program to implement a shift-reduce parser for simple expressions. Trace the parsing steps manually or using a parse table.
7. Modify your parser to construct and print the abstract syntax tree (AST). Display the tree in pre-order or in graphical form.
8. Generate three-address code (TAC) for assignment statements, conditional expressions, and arithmetic expressions. Implement backpatching for boolean expressions.
9. Identify basic blocks from a sequence of TAC instructions. Construct DAG for each basic block to eliminate common sub-expressions.
10. Translate the DAG to target code using simple register allocation techniques. Generate assembly-like code for the basic block.

Teaching Scheme		Semester VI Computer Networks Laboratory	Examination Scheme	
TH	-		CA	40
PR	2		MSE	–
CR	1		ESE	60

List of Experiments:

1. Simulate and Understand IP forwarding within a LAN and across a router.
2. Study the working of spanning tree algorithm by varying the priority among the switches.
3. Understand the working of "Connection Establishment" in TCP using a network simulator.
4. Study how the Data Rate of a Wireless LAN (IEEE 802.11b) network varies as the distance between the Access Point and the wireless nodes is varied.
5. Study the working and routing table formation of Interior routing protocols, i.e. Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).
6. Plot the characteristic curve throughput versus offered traffic for a Slotted ALOHA system.
7. Understand the impact of bit error rate on packet error and investigate the impact of error of a simple hub based CSMA / CD network.
8. Study the performance of networks based on Star, Bus and Ring topologies.
9. TCP/IP Sockets: Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
10. Write a program for calculating the shortest oath using Link State Routing Algorithms.

Teaching Scheme		Semester VI Software Engineering Laboratory	Examination Scheme	
TH	-		CA	40
PR	2		MSE	–
CR	1		ESE	60

List of Experiments:

1. To perform the system analysis: Requirement analysis, SRS. (Both Functional and Nonfunctional requirements. For a set of 10 sample problems, from a book on Software Engineering by Rajib Mall.)
2. To perform the function oriented diagram: DFD and Structured chart.
3. To perform the user‘s view analysis: Use case diagram.
4. To draw the structural view diagram: Class diagram, object diagram.
5. To draw the behavioral view diagram: Sequence diagram, Collaboration diagram.
6. To draw the behavioral view diagram: State-chart diagram, Activity diagram.
7. To draw the implementation view diagram: Component diagram.
8. To draw the environmental view diagram: Deployment diagram.
9. To perform various testing using the testing tool unit testing, integration testing

Teaching Scheme		Semester VI Mini Project	Examination Scheme	
TH	-		CA	40
PR	2		MSE	–
CR	1		ESE	60

Guidelines:

The students shall study in group of two members (or individual) on some special topic beyond the scope of the syllabus under the subjects of Artificial Intelligence, Data Science, Electronics Engineering and Computer Science Engineering or inter discipline branch from current literature, by referring the current technical journal or reference books, under the guidance of the teacher.
In this subject head, it is expected that the student should complete the following tasks.

1. Identify problem statement / idea which is solving one problem preferably local problem may bein their University / College / nearby vicinity.
2. Do the literature survey,
3. Design the solutions
4. Implement solution using latest technology
5. Write 20-25 pages report (use of latex is more suitable).
6. Present / demonstrate the solution in front of faculty member

The students shall prepare his report and execution of project for other students of his class in the presence of his guide and examiner. The student is permitted to use audio-visual aids or any other such teaching aids.

Continues Assessment:
The Continues Assessment for this head will consists of the report written in a technical reporting manner and execution of project will be assessed by the internal examiner appointed by the Head of concern department of the institution.