P.A. COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
POLLACHI - 642 002
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(Accredited by NBA)

CURRICULA AND SYLLABI
M.E. CSE PROGRAMME
REGULATION 2019

P.A. COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
POLLACHI - 642 002
REGULATION 2019
M.E. COMPUTER SCIENCE AND ENGINEERING
CURRICULUM AND SYLLABUS

SEMESTER I

S.No.	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	C
THEORY							
1	19CSFC101	Research Methodology and IPR	FC	3	0	0	3
2	19CSFC102	Mathematical Foundations of Computer Science	FC	3	1	0	4
3	19CSPC103	Advanced Data Structures	PC	3	0	0	3
4	19CSPC104	Soft Computing	PC	3	0	0	3
5	19CSPE1xx	Professional Elective I	PE	3	0	0	3
PRACTICALS							
6	19CSPC105	Advanced Data Structures Laboratory	PC	0	0	3	1.5
7	19CSPC106	Soft Computing Laboratory	PC	0	0	3	1.5
			TOTAL	**15**	**1**	**6**	**19**

SEMESTER II

S.No.	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	C
THEORY							
1.	19CSPC201	Advanced Database Systems	PC	3	0	1	4
2.	19CSPC202	Network Design and Technologies	PC	3	0	0	3
3.	19CSPC203	Big Data Analytics	PC	3	0	0	3
4.	19CSPE2xx	Professional Elective II	PE	3	0	0	3
5.	19CSPE2xx	Professional Elective III	PE	3	0	0	3
PRACTICALS							
6.	19CSPC204	Networks Laboratory	PC	0	0	3	1.5
7.	19CSPC205	Big Data Analytics Laboratory	PC	0	0	3	1.5
			TOTAL	**15**	**0**	**7**	**19**

SEMESTER III

S.No.	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	C
THEORY							
1.	19CSPE3xx	Professional Elective IV	PE	3	0	0	3
2.		Open Elective I	OE	3	0	0	3
PRACTICALS							
3.	19CSEE301	Project Phase I	EEC	0	0	20	10
			TOTAL	**6**	**0**	**20**	**16**

SEMESTER IV

S.NO.	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	C
PRACTICALS							
1.	19CSEE401	Project Phase II	EEC	0	0	32	**16**
			TOTAL	**0**	**0**	**32**	**16**

LIST OF PROFESSIONAL ELECTIVES - I							
S. NO	**COURSE CODE**	**COURSE TITLE**	**CATEGORY**	**L**	**T**	**P**	**C**
1.	19CSPE101	Virtual Reality and Augmented Reality	PE	3	0	0	3
2.	19CSPE102	Principles of Programming Languages	PE	3	0	0	3
3.	19CSPE103	Privacy and Security in IoT	PE	3	0	0	3
4.	19CSPE104	Embedded Systems	PE	3	0	0	3
5.	19CSPE105	Advanced Computer Architecture	PE	3	0	0	3
6.	19CSPE106	Web Engineering	PE	3	0	0	3
7.	19CSPE107	Deep Learning	PE	3	0	0	3
8.	19CSPE108	Wireless Sensor Networks	PE	3	0	0	3
9.	19CSPE109	Introduction to Intelligent Systems	PE	3	0	0	3

LIST OF PROFESSIONAL ELECTIVES - II							
S. NO	**COURSE CODE**	**COURSE TITLE**	**CATEGORY**	**L**	**T**	**P**	**C**
1.	19CSPE201	Data Science	PE	3	0	0	3
2.	19CSPE202	Advanced Distributed Systems	PE	3	0	0	3
3.	19CSPE203	Artificial Neural Networks	PE	3	0	0	3
4.	19CSPE204	Theory of Modern Compilers	PE	3	0	0	3
5.	19CSPE205	Mobile and Pervasive Computing	PE	3	0	0	3
6.	19CSPE206	Parallel Programming Paradigms	PE	3	0	0	3
7.	19CSPE207	Information Storage Management	PE	3	0	0	3
8.	19CSPE208	Advanced Wireless and Mobile Networks	PE	3	0	0	3
9.	19CSPE209	Software Architectures and Design	PE	3	0	0	3

LIST OF PROFESSIONAL ELECTIVES - III							
S. NO	**COURSE CODE**	**COURSE TITLE**	**CATEGORY**	**L**	**T**	**P**	**C**
1.	19CSPE210	Quantum Computing	PE	3	0	0	3
2.	19CSPE211	Real Time Operating Systems	PE	3	0	0	3
3.	19CSPE212	Software Quality Assurance and Testing	PE	3	0	0	3

4.	19CSPE213	Advanced Microcontrollers and Applications in Embedded systems.	PE	3	0	0	3
5.	19CSPE214	Social Networks and Analysis	PE	3	0	0	3
6.	19CSPE215	Data Preparation and Analysis	PE	3	0	0	3
7.	19CSPE216	Secure Software Design & Enterprise Computing	PE	3	0	0	3
8.	19CSPE217	Performance Analysis of Computer Systems	PE	3	0	0	3
9.	19CSPE218	Multi-core Architectures and Programming	PE	3	0	0	3

LIST OF PROFESSIONAL ELECTIVES - IV

S. NO	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	C
1.	19CSPE301	Cyber security and Digital forensics	PE	3	0	0	3
2.	19CSPE302	Open Source Software	PE	3	0	0	3
3.	19CSPE303	Block Chain Technology	PE	3	0	0	3
4.	19CSPE304	Robotics Process Automation	PE	3	0	0	3
5.	19CSPE305	Advanced Software Engineering	PE	3	0	0	3
6.	19CSPE306	GPU Computing	PE	3	0	0	3
7.	19CSPE307	Advanced Optimization Techniques	PE	3	0	0	3
8.	19CSPE308	Data Visualization Techniques	PE	3	0	0	3
9.	19CSPE309	Computer Graphics and Multimedia	PE	3	0	0	3

OPEN ELECTIVES

S.NO.	COURSE CODE	COURSE TITLE	CATEGORY	L	T	P	C
1.	19PEOE01	Energy Auditing	OE	3	0	0	3
2.	19PEOE02	Advanced Energy Storage Technology	OE	3	0	0	3
3.	19PEOE03	Virtual Instrumentation	OE	3	0	0	3
4.	19PEOE04	Distribution Automation System	OE	3	0	0	3
5.	19PEOE05	Power Quality Assessment and Mitigation	OE	3	0	0	3

6.	19CSOE06	Component Based System Design	OE	3	0	0	3
7.	19CSOE07	Pattern Recognition	OE	3	0	0	3
8.	19CSOE08	Artificial Intelligence and Machine Learning	OE	3	0	0	3
9.	19CSOE09	Computer Network Engineering	OE	3	0	0	3
10.	19CSOE10	Green Computing	OE	3	0	0	3

CURRICULUM BREAKDOWN STRUCTURE UNDER REGULATION 2019

S.No.	SUBJECT AREA	CREDITS AS PER SEMESTER				CREDITS TOTAL	PERCENTAGE
		I	II	III	IV		
1.	**FC**	7	0	0	0	**07**	**10.00**
2.	**PC**	9	13	0	0	**22**	**31.43**
3.	**PE**	3	6	3	0	**12**	**17.14**
4.	**EEC**	0	0	10	16	**26**	**37.14**
5.	**OE**	0	0	3	0	**03**	**4.29**
	Total	**19**	**19**	**16**	**16**	**70**	**100%**

SEMESTER I

19CSFC101	RESEARCH METHODOLOGY AND IPR	SEMESTER I
		L T P C
		3 0 0 3

COURSE OBJECTIVES:

- To understand definition and objectives of Research.
- To be familiar with Quantitative methods for problem solving.
- To know Data description and report writing.

UNIT-I: INTRODUCTION **9**

Definition and objectives of research – Types of research, Various steps in research process, Mathematical tools for analysis, developing a research question – Choice of a problem Literature review, Surveying, synthesizing, Critical analysis, Reading materials, Reviewing, Rethinking, Critical evaluation, Interpretation, Research Purposes, Ethics in research – APA Ethics code.

UNIT-II: QUANTITATIVE METHODS FOR PROBLEM SOLVING **9**

Statistical modeling and analysis, Time series analysis probability distributions, Fundamentals of statistical analysis and inference, Multivariate methods, Concepts of correlation and regression, Fundamentals of time series analysis and spectral analysis, Error analysis, Applications of spectral analysis.

UNIT-III: DATA DESCRIPTION AND REPORT WRITING **9**

Tabular and graphical description of data: Tables and graphs of frequency data of one variable, Tables and graphs that show the relationship between two variables, Relation between frequency distributions and other graphs, preparing data for analysis.

Structure and components of research report, Types of report, Layout of research report, Mechanism of writing a research report, Referencing in academic writing.

UNIT-IV: INTELLECTUAL PROPERTY **9**

Nature of Intellectual Property: Patents, Designs, Trade and copyright. Process of [patenting and development: Technological research, Innovation, patenting, Development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

UNIT-V: PATENT RIGHTS **9**

Patent rights: Scope of Patent rights. Licensing and transfer of technology. Patent information and databases. Geographical indications.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Stuart Melville and Wayne Goddard, "Research Methodology: An Introduction for Science and Engineering Students", Juta Academic, 1996.
2. Donald H. McBurney and Theresa White, "Research Methods", 9th Edition, Cengage Learning, 2013.
3. Ranjit Kumar, "Research Methodology: A Step by Step Guide for Beginners", 4th Edition, 2014.

4. Dr. Kotharia. C.R and Gaurav Garg, "Research Methodology: Methods and Trends", New age international publishers, 3rd Edition, 2014.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Develop research question.
CO2: Perform exhaustive literature survey.
CO3: Apply right problem solving methods.
CO4: Prepare data for analysis.
CO5: Write research report.

19CSFC102	MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE	SEMESTER I			
		L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

- To know random variables and discrete and continuous distributions.
- To understand Queuing models.
- To acquire knowledge on tests of sampling.
- To understand correlation and regression analysis.

UNIT-I: RANDOM VARIABLES **9+3**

Random variables – Binomial, Geometric, Poisson, Uniform, Exponential, Erlang and Normal distributions – Functions of a random variable – Moments and Moment generating function.

UNIT-II: MARKOVIAN QUEUEING MODELS **9+3**

Markovian models – Birth and Death Queuing models – Steady state results: Single and multiple server queuing models – Queue with finite waiting rooms – Finite source – Finite source models – Little's formula.

UNIT-III: NON-MARKOVIAN QUEUES AND QUEUE NETWORKS **9+3**

M/G/1 queue – Pollazack-Khintchine formula, Series queues – Open and closed networks.

UNIT-IV: TESTING OF HYPOTHESIS **9+3**

Sampling distributions – Estimation of parameters – Statistical hypothesis – Tests based on Normal, t, Chi Square and F distributions for mean, Variance and proportion.

UNIT-V: CORRELATION AND REGRESSION ANALYSIS **9+3**

Coefficient of correlation – Rank correlation – Regression lines – Multiple and Partial correlation – Partial regression – Regression planes (Problems only).

Contact Periods:
Lecture: 45 Periods Tutorial: 15 Periods Practical: 0 Periods Total: 60 Periods

REFERENCES

1. Veerarajan T., "Probability and Random Processes (with Queueing Theory and Queueing Networks)", 4th Edition , McGraw Hill Education(India)Pvt Ltd., New Delhi, 2016.
2. Medhi J., "Introduction to Queuing Systems and applications", 1st Edition, New Age International(P) Ltd,New Delhi, 2015.
3. Gross D and Harris C.M., "Fundamentals of Queuing theory", John Wiley and Sons, New York, 1998.
4. Gupta S.C and Kapoor V.K., "Fundamentals of Mathematical Statistics", Sultan Chand & Sons, New Delhi, 2015.
5. Gupta S.P., "Statistical Methods", Sultan Chand & Sons, New Delhi, 2015.
6. Veerarajan T., "Higher Engineering Mathematics", Yes Dee Publishing Pvt Ltd, Chennai, 2016.
7. Kandasamy P, Thilagavathy K and Gunavathy K., "Probability and Queueing Theory", S. Chand & Co, Ramnagar, New Delhi, Reprint 2013.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain random variables and their distributions and also moments and moment generating functions for their mean and variance.

CO2: Explain probable values of queues with single and multi-server models.

CO3: Explain tests of sampling for large and small samples.

CO4: Explain probability distributions of discrete and continuous random variables.

CO5: Calculate coefficient of correlation, regression coefficients, multiple and partial correlation including regression plane.

19CSPC103 ADVANCED DATA STRUCTURES SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- The student should be able to choose appropriate data structures, understand the ADT/ libraries and use it to design algorithms for a specific problem.
- Students should be able to understand the necessary mathematical abstraction to solve problems.
- To familiarize students with advanced paradigms and data structure used to solve algorithmic problems.
- Student should be able to come up with analysis of efficiency and proofs of correctness.

UNIT-I: DICTIONARIES AND HASHING 9

Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries. Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic, Recent Trends in Hashing, Trees and various computational geometry methods for efficiently solving the new evolving problem.

UNIT-II: HIERARCHICAL DATA STRUCTURES 9

Binary Search Trees: Basics – Querying a binary search tree – Insertion and deletion- Red-Black trees: Properties of Red-Black Trees – Rotations – Insertion – Deletion – B-Trees: Definition of B-trees – Basic operations on B-Trees – Deleting a key from a B-Tree – Fibonacci heaps: structure – Mergeable – Heap operations – Decreasing a key and deleting a Node – Bounding the maximum degree.

UNIT-III: GRAPHS 9

Elementary graph algorithms: Representations of graphs – Breadth-first search – Depth-first search – Topological sort – Strongly connected components – Minimum spanning trees: Growing a minimum spanning tree – Kruskal and prim- Single-source shortest paths: The Bellman-ford algorithm – Single – Source shortest paths in directed acyclic graphs – Dijkstra‘s algorithm; All – pairs shortest paths: Shortest paths and matrix multiplication – The Floyd – Warshall algorithm.

UNIT-IV: TEXT PROCESSING 9

Sting operations – Brute-force pattern matching – The Boyer-Moore algorithm – The Knuth-Morris-Pratt algorithm – Standard tries – Compressed tries – Suffix tries – The Huffman Coding algorithm –The Longest Common Subsequence Problem (LCS) – Applying dynamic programming to the LCS problem.

UNIT-V: COMPUTATIONAL GEOMETRY 9

One dimensional range searching – Two dimensional range searching – Constructing a priority search tree – Searching a priority search tree – Priority range trees – Quad trees – k-D trees.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition, Pearson,

2007.
2. M T Goodrich, Roberto Tamassia, “Algorithm Design”, John Wiley, 2002.
3. Alfred V.Aho, John E.Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Cognize the implementation of symbol table using hashing techniques.

CO2: Develop and analyze algorithms for red-black trees, B-trees and Splay trees.

CO3: Develop and analyze algorithms for elementary graph algorithms.

CO4: Develop algorithms for text processing applications.

CO5: Identify suitable data structures and develop algorithms for computational geometry problems.

19CSPC104 **SOFT COMPUTING** **SEMESTER I**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand classifier of Neutral network.
- To know Fuzzy sets and rules.
- To understand Neuro Fuzzy modelling techniques.
- To understand Genetic algorithms.
- To Acquire knowledge on Integration of hybrid systems.

UNIT-I: NEURAL NETWORKS **9**

Supervised learning Neural networks – Perceptrons – Adaline - Back propagation – Multilayer perceptrons – Radial basis function networks – Unsupervised learning and other Neural networks – Competitive learning networks – Kohonen self-organizing networks – Learning vector quantization – Hebbian learning.

UNIT-II: FUZZY LOGIC SYSTEM **9**

Fuzzy sets – Basic definition and terminology – Set-theoretic operations – Member function – Fuzzy rules and Fuzzy reasoning – Extension principle and Fuzzy relations – Fuzzy If-Then rules – Fuzzy reasoning – Fuzzy inference systems – Mamdani Fuzzy models –Sugeno Fuzzy models – Defuzzification.

UNIT-III: NEURO FUZZY MODELING **9**

Adaptive Neuro – Fuzzy inference systems – Architecture – Hybrid learning algorithm –learning methods that cross – fertilize ANFIS and RBFN-Coactive Neuro-Fuzzy modeling – Framework – Neuron functions for Adaptive networks – Neuro Fuzzy spectrum.

UNIT-IV: GENETIC ALGORITHMS **9**

Traditional optimization and search methods – Simple genetic algorithm – Reproduction – Crossover – Mutation – Schemata – Schema theorem – Two and K-arm Bandit problem – Improvements in basic techniques – Selection schemes – Scaling mechanisms – Ranking procedures.

UNIT-V: HYBRID SYSTEMS **9**

Integration of neural networks – fuzzy logic and genetic algorithms.

Contact Periods:

Lecture: 45 Periods **Tutorial: 0 Periods** **Practical: 0 Periods** **Total: 45 Periods**

REFERENCES

1. Jang J.S.R, Sun C.T and Mizutani E., "Neuro - Fuzzy and Soft Computing ", Pearson Education, 2009.
2. Timothy Ross J., "Fuzzy Logic with Engineering Applications", John Wiley and Sons Pvt.Ltd., 2010.
3. Zimmermann H.J., "Fuzzy Set Theory and its Applications", 4th Edition, Kluwer Academic Publishers, 2013.
4. Davis E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning" Addison Wesley, N.Y., 1989.

5. Rajasekaran S and Pai. G.A.V., “Neural Networks, Fuzzy Logic and Genetic Algorithms”, PHI, 2003.
6. Elaine Rich and Kevin Knight, “Artificial Intelligence”, 3rd Edition, Tata McGraw Hill, 2011.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Analyze various neural network architectures.
CO2: Analyze the ideas of Neural networks, fuzzy logic and use of heuristics
CO3: Explain Fuzzy sets and rules.
CO4: Analyze and gain insight onto Neuro Fuzzy modeling and control.
CO5: Analyze the genetic algorithms and their applications.

19CSPC105 ADVANCED DATA STRUCTURES LABORATORY SEMESTER I

L	T	P	C
0	0	3	1.5

COURSE OBJECTIVES:

- To acquire the knowledge of using advanced tree structures.
- To learn the usage of heap structures.
- To understand the usage of graph structures and spanning trees.

LIST OF EXPERIMENTS:

Each student has to work individually on assigned lab exercises. Lab sessions could be scheduled as one contiguous four-hour session per week or two two-hour sessions per week. There will be about 15 exercises in a semester. It is recommended that all implementations are carried out in Java. If C or C++ has to be used, then the threads library will be required for concurrency. Exercises should be designed to cover the following topics:

1. Implementation of Merge sort and Quick sort-analysis
2. Implementation of a Binary search tree
3. Red-Black tree implementation
4. Heap implementation
5. Fibonacci Heap implementation
6. Graph traversals
7. Spanning tree implementation
8. Shortest path algorithms (Dijkstra's algorithm, Bellmann Ford Algorithm)
9. Implementation of Matrix Chain multiplication
10. Activity selection and Huffman coding implementation

Contact Periods:

Lecture: 0 Periods Tutorial: 0 Periods Practical: 45 Periods Total: 45 Periods

REFERENCES

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 2nd Edition, Pearson, 2004.
2. Goodrich. M.T and Roberto Tamassia, "Algorithm Design", John Wiley, 2002.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Design and implement basic and advanced data structures extensively.

CO2: Design algorithms using graph structures.

CO3: Design and develop efficient algorithms with minimum complexity using design techniques.

19CSPC106 SOFT COMPUTING LABORATORY SEMESTER I

L	T	P	C
0	0	3	1.5

COURSE OBJECTIVES:

- To implement map reduce programs for processing big data.
- To realize storage of big data using H base, Mongo DB.
- To analyze big data using linear models.
- To analyze big data using machine learning techniques such as SVM/Decision tree. classification and clustering.

LIST OF EXPERIMENTS:

1. Create a perceptron with appropriate no. of inputs and outputs. Train it using fixed increment learning algorithm until no change in weights is required. Output the final weights
2. Create a simple ADALINE network with appropriate no. of input and output nodes. Train it using delta learning rule until no change in weights is required. Output the final weights. Train the auto correlator by given patterns: A1=(-1,1,-1,1), A2=(1,1,1,-1), A3=(-1, -1, - 1, 1). Test it using patterns: Ax=(-1,1,-1,1), Ay=(1,1,1,1), Az=(-1,-1,-1,-1)
3. Train the hetrocorrelator using multiple training encoding strategy for given patterns: A1=(000111001) B1=(010000111), A2=(111001110) B2=(100000001), A3=(110110101) B3(101001010). Test it using pattern A2
4. Implement union, Intersection, Complement and difference operations on Fuzzy sets. Also create Fuzzy relation by cartesian product of any two Fuzzy sets and perform maxmin composition on any two Fuzzy relations
5. Solve Greg Viot's Fuzzy cruise controller using MATLAB Fuzzy logic toolbox
6. Solve Air conditioner controller using MATLAB Fuzzy logic toolbox
7. Implement TSP using GA

Contact Periods:
Lecture: 0 Periods Tutorial: 0 Periods Practical: 45 Periods Total: 45 Periods

REFERENCES

1. Rajasekaran S and Vijayalakshami G.A., "Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications", PHI.
2. Davis E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning" Addison Wesley, N.Y., 1989.
3. Chin Teng Lin, George Lee.C.S., "Neuro-Fuzzy Systems", PHI.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Process big data using Hadoop framework.
CO2: Build and apply linear and logistic regression models.
CO3: Perform data analysis with machine learning methods and Perform graphical data analysis.

SEMESTER II

19CSPC201	ADVANCED DATABASE SYSTEMS	SEMESTER II			
		L	T	P	C
		3	0	1	4

COURSE OBJECTIVES:

- To design data base with ER model and relational model.
- To understand data storage and retrieval techniques.
- To perform Query processing and transaction management.
- To understand parallel and distributed databases.
- To understand enhanced data models.
- To know NoSQL databases.

UNIT-I: DATABASE DESIGN **9**

Data models – ER model: Constraints – ER-diagrams – Extended ER features – Relational database design: Good relational designs – Normal forms – Functional dependencies – Decomposition algorithms – Modeling temporal data – Application design and development – Performance tuning.

UNIT-II: STORAGE, QUERYING AND TRANSACTION MANAGEMENT **9**

Indexing and hashing – Query processing and optimization – Transaction management: Concurrency and recovery – Advanced transaction processing.

UNIT-III: PARALLEL AND DISTRIBUTED DATABASES **9**

Database system architecture – Parallel databases: Parallelism, Query optimization and design of parallel system – Distributed databases: Distributed storage and transactions, Concurrency control, Query processing, Cloud based databases, Directory systems.

UNIT-IV: DATABASE SECURITY AND ENHANCED DATA MODELS **9**

Database security: Issues – Access control mechanisms – SQL injection – Statistical database security – Advanced data models: Active database – Temporal database– Multimedia database – Spatial and deductive databases – XML.

UNIT-V: HYBRID SYSTEMS **9**

Emergence – Aggregate data models – Distribution models – Consistency – Key value databases – Document databases – Column family stores – Graph databases – Schema migration – Polyglot persistence.

List of Experiments: 15

1. Database Design and Normalization
2. Accessing Databases from Programs using JDBC
3. Building Web Applications using PHP & MySQL
4. Indexing and Query Processing
5. Concurrency and Transactions
6. Big Data Analytics using Hadoop

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 15 Periods Total: 60 Periods

REFERENCES

1. Abraham Silberschatz, Henry F. Korth and Sudarshan S., "Database System Concepts", 6th Edition, McGraw-Hill, 2012.
2. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled - A Brief Guide to the Emerging World of Polyglot Persistence", Pearson Education, 2013.
3. Elmasri R and Navethe S., "Fundamentals of Database Systems", 7th Edition, Pearson Education.
4. Raghu Ramakrishnan and Gehrke, "Database Management Systems", 3rd Edition, McGraw Hill, 2003.
5. Thomas Cannoly and Carolyn Begg, "Database Systems, A Practical Approach to Design, Implementation and Management" Addison- Wesley Professional, 2012.
6. Tamer Ozsu M and Patrick Valdurriez, "Principles of Distributed Database Systems", 3rd Edition, Springer, 2011.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Design and develop a relational data model.

CO2: Express the storage and data access mechanisms.

CO3: Perform query optimization and use transaction management techniques.

CO4: Apply concurrency control and query optimization algorithms in parallel and distributed data models.

CO5: Use enhanced data models and use NoSQL databases.

19CSPC202 NETWORK DESIGN AND TECHNOLOGIES SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand the principles required for network design.
- To explore various technologies in the wireless domain.
- To study about 3G and 4G cellular networks.
- To understand the paradigm of Software defined networks.

UNIT-I: NETWORK DESIGN 9

Advanced multiplexing – Code division multiplexing – DWDM and OFDM – Shared media networks – Switched networks – End to end semantics – Connectionless – Connection oriented – Wireless Scenarios – Applications – Quality of service – End to end level and network level solutions. LAN cabling topologies – Ethernet switches – Routers – Firewalls and L3 switches – Remote access technologies and devices – Modems and DSLs – SLIP and PPP – Core networks and distribution networks.

UNIT-II: WIRELESS NETWORKS 9

IEEE802.16 and WiMAX – Security – Advanced 802.16 functionalities – Mobile WiMAX - 802.16e – Network infrastructure – WLAN – Configuration – Management operation – Security – IEEE 802.11e and WMM – QoS – Comparison of WLAN and UMTS – Bluetooth – Protocol Stack – Security – Profiles.

UNIT-III: CELLULAR NETWORKS 9

GSM – Mobility management and call control – GPRS – Network elements – Radio resource management – Mobility management and session management – Small screen Web browsing over GPRS and EDGE – MMS over GPRS – UMTS – Channel structure on the air interface – UTRAN – Core and radio network mobility management – UMTS security.

UNIT-IV: 4G NETWORKS 9

LTE – Network architecture and interfaces – FDD air interface and radio networks – Scheduling – Mobility management and power optimization – LTE security architecture – Interconnection with UMTS and GSM – LTE advanced (3GPPP Release 10) – 4G networks and composite radio Environment – Protocol boosters – Hybrid 4G Wireless networks protocols – Green wireless networks – Physical layer and multiple access – Channel modelling for 4G – Introduction to 5G.

UNIT-V: SOFTWARE DEFINED SYSTEMS 9

Introduction – Centralized and distributed control and data planes – Open flow – SDN controllers – General concepts – VLANs – NVGRE – Open flow – Network overlays – Types – Virtualization – Data plane – I/O – Design of SDN framework.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Erik Dahlman, Stefan Parkvall and Johan Skold, “4G: LTE/LTE-Advanced for Mobile Broadband”, Academic Press, 2013.
2. Jonathan Rodriguez, “Fundamentals of 5G Mobile Networks”, Wiley, 2015.

3. Larry Peterson and Bruce Davie, "Computer Networks: A Systems Approach", 5th Edition, Morgan Kauffman, 2011.
4. Martin Sauter, "From GSM to LTE, An Introduction to Mobile Networks and Mobile Broadband", Wiley, 2014.
5. Martin Sauter, "Beyond 3G - Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0", Wiley, 2009.
6. Naveen Chilamkurti, Sherali Zeadally and Hakima Chaouchi, "Next-Generation Wireless Technologies", Springer, 2013.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Identify the components required for designing a network.
CO2: Design a network at a high-level using different networking technologies.
CO3: Analyze the various protocols of wireless and cellular networks.
CO4: Discuss the features of 4G and 5G networks.
CO5: Experiment with software defined networks.

19CSPC203	BIG DATA ANALYTICS	SEMESTER II			
		L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- To understand statistical methods.
- To implement Bayesian, support vector and Kernel methods.
- To perform time series analysis and rule induction.
- To understand Neural networks and Fuzzy logic.
- To understand visualization techniques.

UNIT-I: STATISTICAL CONCEPTS AND METHODS 9

Statistical concepts: Probability, Sampling and sampling distributions, Statistical inference, prediction and Prediction errors – Resampling – Statistical method: Linear models, Regression modeling, Multivariate analysis.

UNIT-II: BAYESIAN METHODS AND SUPPORT VECTOR AND KERNEL METHODS 9

Bayesian methods: Bayesian paradigm, Modeling, Inference and networks – Support vector and Kernel methods: Kernel perceptron, Overfitting and generalization bounds, Support vector machines, Kernel PCA and CCA.

UNIT-III: TIME SERIES ANALYSIS AND RULE INDUCTION 9

Analysis of time series: Linear systems analysis, Nonlinear dynamics, Delay coordinate embedding – Rule induction: Propositional rule learning, Rule learning as search, Evaluating quality of rules, Propositional rule induction, First order rules – ILP systems.

UNIT-IV: NEURAL NETWORKS AND FUZZY LOGIC 9

Neural networks: Learning and generalization, Competitive learning, Principal component analysis and Neural networks; Fuzzy logic: Extracting Fuzzy models from data, Fuzzy decision trees.

UNIT-V: STOCHASTIC SEARCH METHODS AND VISUALIZATION 9

Stochastic Search methods: Stochastic search by simulated annealing, Adaptive search by evolution – Evolution strategies – Genetic algorithms & programming – Visualization: Classification of visual data analysis techniques, Data type to be visualized, Visualization techniques, Interaction techniques and specific visual data analysis techniques.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Michael Berthold and David J. Hand, "Intelligent Data Analysis-An Introduction", 2nd Edition, Springer, 2007.
2. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with advanced analytics", John Wiley & Sons, 2012.
3. Jimmy Lin and Chris Dyer, "Data Intensive Text Processing using Map Reduce", Morgan and Claypool Publishers, 2010.
4. Tom White, "Hadoop: The Definitive Guide", O`Reilly Publishers, 2012.

5. David Loshin, “Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph”, Morgan Kaufmann, 2013.
6. Paul Zikopoulos and Chris Eaton, “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”, McGraw-Hill Education, 2011.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the statistical concepts and methods.

CO2: Recognize Bayesian, support vector and Kernel methods.

CO3: Perform time series analysis.

CO4: Use rule induction. Apply Neural network and Fuzzy logic.

CO5: Demonstrate Stochastic search methods. Visualization techniques.

19CSPC204 NETWORKS LABORATORY SEMESTER II

L	T	P	C
0	0	3	1.5

COURSE OBJECTIVES:

- To acquire the knowledge of designing networks.
- To learn about various network protocols.
- To understand the paradigm of Software defined networks.

LIST OF EXPERIMENTS:

Student will be introduced to the network modeling and simulation, and they will have the opportunity to build some simple networking models using the tool and perform simulations that will help them evaluate their design approaches and expected network performance.

1. **Establishing a Local Area Network (LAN):**
   The main objective is to set up a Local Area Network, concepts involved in this network are IP addressing and the Address Resolution Protocol (ARP). The required equipment's are 192.168.1.1 ,192.168.1.2, 192.168.1.3, Host A Host B Host C, Switch/HUB, three PC`s equipped with at least one NIC, one HUB or Switch and the necessary cables. Once the physical LAN is set up the hosts need to be configured using the ifconfig command. To verify communication among the machines the ping command is used. Next, to manipulate the routing tables at the hosts to understand how machines know where to send packets. Since the ifconfig command places a default route into the routing tables this route must be deleted. to blindfoldˇ the machine. The ping command is used again to show that communication is no longer available. To re-establish communication the routes are put back into the routing table one host at a time. Communication is once again verified using the ping command.

2. **Connecting two LANs using multi-router topology with static routes:**
   The main objective is to extend routing connection by using multiple routers. The concepts include IP addressing and basic network routing principles. Connect two LANs topology. During router configuration attention is paid to the types of interfaces as additional issues are involved with set- up. For example, the serial interfaces require clocking mechanisms to be set correctly. Once the interfaces are working the ping command is used to check for communication between LANs. The failure of communication illustrates the need for routes to be established inside the routing infrastructure. Static routes are used to show how packets can be transported through any reasonable route. It is run trace route on two different configurations to demonstrate the implementation of different routes. Analyzing the performance of various configurations and protocols. Original TCP versus the above modified one: To compare the performance between the operation of TCP with congestion control and the operation of TCP as implemented. The main objective is for students to examine how TCP responds to a congested network. The concepts involved in the lab include network congestion and the host responsibilities for communicating over a network. This lab requires three PC's connected to a switch. One PC is designated as the target host and the other two PC's will transfer a file from the target host using FTP. A load is placed on the network to simulate congestion and the file is transferred, first by the host using the normal TCP and then by the host using the modified version. This procedure is performed multiple times to determine average statistics. The students are then asked to summarize the results and draw conclusions about the performance differences and the underlying implications for hosts operating in a network environment.

3. **RIP and OSPF Redistribution**

This case study addresses the issue of integrating Routing Information Protocol (RIP) networks with Open Shortest Path First (OSPF) networks. Most OSPF networks also use RIP to communicate with hosts or to communicate with portions of the internetwork that do not use OSPF. This case study should provide examples of how to complete the following phases in redistributing information between RIP and OSPF networks, including the following topics:

- Configuring a RIP Network
- Adding OSPF to the Center of a RIP Network
- Adding OSPF Areas
- Setting Up Mutual Redistribution

4. **Network Security**

This case study should provide the specific actions you can take to improve the security of your network. Before going into specifics, however, you should understand the following basic concepts that are essential to any security system:

- Know your enemy
- Count the cost
- Know your weaknesses
- Limit the scope of access

5. **Controlling Traffic Flow**

In this case study, the firewall router allows incoming new connections to one or more communication servers or hosts. Having a designated router act as a firewall is desirable because it clearly identifies the router's purpose as the external gateway and avoids encumbering other routers with this task. In the event that the internal network needs to isolate itself, the firewall router provides the point of isolation so that the rest of the internal network structure is not affected. Connections to the hosts are restricted to incoming file transfer protocol (FTP) requests and email services. The incoming Telnet, or modem connections to the communication server are screened by the communication server running TACACS username authentication.

6. **Defining Access Lists**

Access lists define the actual traffic that will be permitted or denied, whereas an access group applies an access list definition to an interface. Access lists can be used to deny connections that are known to be a security risk and then permit all other connections, or to permit those connections that are considered acceptable and deny all the rest. For firewall implementation, the latter is the more secure method. In this case study, incoming email and news are permitted for a few hosts, but FTP, Telnet, and rlogin services are permitted only to hosts on the firewall subnet. IP extended access lists (range 100 to 199) and transmission control protocol (TCP) or user datagram protocol (UDP) port numbers are used to filter traffic. When a connection is to be established for email, Telnet, FTP, and so forth, the connection will attempt to open a service on a specified port number. You can, therefore, filter out selected types of connections by denying packets that are attempting to use that service. An access list is invoked after a routing decision has been made but before the packet is sent out on an interface. The best place to define an access list is on a preferred host using your favorite text editor. You can create a file that contains the access-list commands, place the file (marked readable) in the default TFTP directory, and then network load the file onto the router.

7. **Configuring a fire wall**
   Consider a Fire wall communication server with single inbound modem. Configure the modem to ensure security for LAN.

8. **Integrating EIGRP (Enhanced Interior Gateway Routing Protocol) into Existing Networks:**
   The case study should provide the benefits and considerations involved in integrating Enhanced IGRP into the following types of internetworks:
   • IP—The existing IP network is running IGRP
   • Novell IPX—The existing IPX network is running RIP and SAP
   • AppleTalk—The existing AppleTalk network is running the Routing Table Maintenance **Protocol (RTMP)**
   When integrating Enhanced IGRP into existing networks, plan a phased implementation. Add Enhanced IGRP at the periphery of the network by configuring Enhanced IGRP on a boundary router on the backbone off the core network. Then integrate Enhanced IGRP into the core.

Contact Periods:
Lecture: 0 Periods Tutorial: 0 Periods Practical: 45 Periods Total: 45 Periods

REFERENCES

1. Erik Dahlman, Stefan Parkvall and Johan Skold, "4G: LTE/LTE-Advanced for Mobile Broadband", Academic Press, 2013.
2. Jonathan Rodriguez, "Fundamentals of 5G Mobile Networks", Wiley, 2015.
3. Larry Peterson and Bruce Davie, "Computer Networks: A Systems Approach", 5th edition, Morgan Kauffman, 2011.
4. Martin Sauter, "From GSM to LTE, An Introduction to Mobile Networks and Mobile Broadband", Wiley, 2014.
5. Martin Sauter, "Beyond 3G - Bringing Networks, Terminals and the Web Together: LTE, WiMAX, IMS, 4G Devices and the Mobile Web 2.0", Wiley, 2009.
6. Naveen Chilamkurti, Sherali Zeadally and Hakima Chaouchi, "Next-Generation Wireless Technologies", Springer, 2013.

COURSE OUTCOMES:
Upon successful completion of the course, students should be able to:
CO1: Design a network at high-level using different networking technologies.
CO2: Analyze the various protocols of wireless and cellular networks.
CO3: Experiment with software defined networks.

19CSPC205 BIG DATA ANALYTICS LABORATORY SEMESTER II

L	T	P	C
0	0	3	1.5

COURSE OBJECTIVES:

- To implement Map Reduce programs for processing big data.
- To realize storage of big data using H base, Mongo DB.
- To analyze big data using linear models.
- To analyze big data using machine learning techniques such as SVM / Decision tree classification and clustering.

LIST OF EXPERIMENTS:

Hadoop

1. Install, configure and run Hadoop and HDFS
2. Implement word count / frequency programs using Map Reduce
3. Implement an MR program that processes a weather dataset

R

4. Implement Linear and logistic Regression
5. Implement SVM / Decision tree classification techniques
6. Implement clustering techniques
7. Visualize data using any plotting framework
8. Implement an application that stores big data in Hbase / MongoDB / Pig using Hadoop / R.

Contact Periods:

Lecture: 0 Periods Tutorial: 0 Periods Practical: 45 Periods Total: 45 Periods

REFERENCES

1. Alan Gates and Daniel Dai, "Programming Pig - Dataflow scripting with Hadoop", O'Reilley, 2nd Edition, 2016.
2. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, "An Introduction to Statistical Learning with Applications in R", Springer Publications, 2015(Corrected 6th Printing)
3. Hadley Wickham, "ggplot2 – Elegant Graphics for Data Analysis", Springer Publications, 2nd Edition, 2016.
4. Kristina Chodorow, "Mongo DB: The Definitive Guide – Powerful and Scalable Data Storage", O' Reilley, 2nd Edition, 2013.
5. Lars George, "HBase: The Definitive Guide", O'Reilley, 2015.
6. Tom White, "Hadoop: The Definitive Guide – Storage and Analysis at Internet Scale", O'Reilley, 4th Edition, 2015.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Process big data using Hadoop framework.

CO2: Build and apply linear and logistic regression models.

CO3: Perform data analysis with machine learning methods and Perform graphical data analysis.

PROFESSIONAL ELECTIVES – I

19CSPE101 VIRTUAL REALITY AND AUGMENTED REALITY SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- Ability to understand Geometric modeling.
- Ability to understand virtual environment.
- To acquire knowledge about virtual hardwares and softwares.
- To get introduced to various virtual reality applications.

UNIT-I: INTRODUCTION TO VIRTUAL REALITY 9

Virtual reality & virtual environment : Introduction – Computer graphics – Real time computer graphics – Flight simulation – Virtual environments – requirement – Benefits of virtual reality – Historical development of VR : Introduction – Scientific landmark – 3D Computer Graphics : Introduction – The virtual world space – Positioning the virtual observer – The perspective projection – Human vision – Stereo perspective projection – 3D clipping – Color theory – Simple 3D modeling.– Illumination models – Reflection models – Shading algorithms – Radiosity – Hidden surface removal – Realism – Stereographic image.

UNIT-II: GEOMETRIC MODELLING 9

Geometric Modeling: Introduction – From 2D to 3D – 3D space curves – 3D boundary representation – Geometrical transformations: Introduction – Frames of reference – Modeling transformations – Instances – Picking – Flying – Scaling the VE – Collision detection – A Generic VR system: Introduction – The virtual environment – The Computer environment – VR Technology – Model of interaction – VR systems.

UNIT-III: VIRTUAL ENVIRONMENT 9

Animating the virtual Environment: Introduction – The dynamics of numbers – Linear and Non – linear interpolation – The animation of objects – Linear and nonlinear translation – shape & object inbetweening – Free from deformation – Particle system– Physical Simulation: Introduction – Objects falling in a gravitational field – Rotating wheels – Elastic collisions – Projectiles – Simple pendulum – Springs – Flight dynamics of an aircraft.

UNIT-IV: VR HARDWARES AND SOFTWARES 9

Human factors: Introduction – The eye – The ear – The somatic senses – VR Hardware: Introduction – Sensor hardware – Head – Coupled displays – Acoustic hardware – Integrated VR systems – VR Software: Introduction – Modeling virtual world – Physical simulation – VR toolkits – Introduction to VRML.

UNIT-V: VR APPLICATION 9

Virtual reality applications: Introduction – Engineering – Entertainment – Science – Training – The Future: Introduction – Virtual environments – Modes of interaction.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. John Vince, “Virtual Reality Systems”, Pearson Education Asia, 2007.

2. Gregory C. Burdea & Philippe Coiffet, "Virtual Reality Technology", 2nd Edition, John Wiley & Sons, 2006.
3. Adams, "Visualizations of Virtual Reality", Tata McGraw Hill, 2000.
4. William R. Sherman and Alan B. Craig, "Understanding Virtual Reality: Interface, Application and Design", Morgan Kaufmann, 2008.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the basic concepts of virtual environments.

CO2: Apply geometric modeling and implement 3D interaction techniques.

CO3: Develop immersive virtual reality applications.

CO4: Identify required virtual hardware and software for modeling virtual world.

CO5: Explore different Virtual Reality applications.

19CSPE102 PRINCIPLES OF PROGRAMMING LANGUAGES SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand and describe syntax and semantics of programming languages.
- To understand data, data types, and basic statements.
- To understand call return architecture and ways of implementing.
- To understand object orientation, concurrency and event handling in programming languages.
- To develop programs in non– procedural programming paradigms.

UNIT-I: SYNTAX AND SEMANTICS 9

Evolution of programming languages – describing syntax – Context – Free grammars – attribute grammars – Describing semantics – Lexical analysis – Parsing – Recursive – Decent – Bottom up parsing.

UNIT-II: DATA, DATA TYPES, AND BASIC STATEMENTS 9

Names – Variables – Binding – Type checking – Scope – Scope rules – Lifetime and Garbage collection – Primitive data types – Strings – Array types – Associative arrays – Record types – Union types – Pointers and references – Arithmetic expressions – Overloaded operators – Type conversions – Relational and Boolean expressions – Assignment statements – Mixed mode assignments – Control structures – Selection – Iterations – Branching – Guarded statements.

UNIT-III: SUBPROGRAMS AND IMPLEMENTATIONS 9

Subprograms – Design issues – Local referencing – Parameter passing – Overloaded methods – Generic methods – Design issues for functions – Semantics of Call and return – Implementing simple subprograms – Stack and dynamic local variables – Nested subprograms – Blocks – Dynamic scoping.

UNIT-IV: OBJECT– ORIENTATION, CONCURRENCY, AND EVENT HANDLING 9

Object – Orientation – Design issues For OOP Languages – Implementation of object– Oriented constructs – Concurrency – Semaphores – Monitors – Message passing – Threads – Statement level concurrency – Exception handling – Event handling.

UNIT-V: FUNCTIONAL AND LOGIC PROGRAMMING LANGUAGES 9

Introduction to lambda calculus – Fundamentals of functional programming languages – Programming with Scheme – Programming with ML – Introduction to logic and logic programming – Programming with Prolog – Multi – paradigm languages.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Robert W. Sebesta, "Concepts of Programming Languages", 10th Edition, Addison Wesley, 2012.
2. Michael L. Scott, "Programming Language Pragmatics", 3rd Edition, Morgan Kaufmann, 2009.

3. Kent Dybvig R., “The Scheme programming language”, 4th Edition, MIT Press, 2009.
4. Jeffrey D. Ullman, “Elements of ML programming”, 2nd Edition, Prentice Hall, 1998.
5. Richard A. O'Keefe, “The Craft of Prolog”, MIT Press, 2009.
6. Clocksin W. F and Mellish C.S., “Programming in Prolog: Using the ISO Standard”, 5th Edition, Springer, 2003.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Describe syntax and semantics of programming languages.

CO2: Explain data, data types, and basic statements of programming languages.

CO3: Design and implement subprogram constructs.

CO4: Apply object – oriented, concurrency, and event handling programming constructs.

CO5: Adopt new programming languages.

19CSPE103 PRIVACY AND SECURITY IN IOT SEMESTER I

L	T	P	C
3	**0**	**0**	**3**

COURSE OBJECTIVES:

- Ability to understand the Security requirements in IoT.
- Understand the cryptographic fundamentals for IoT.
- Ability to understand the authentication credentials and access control.
- Understand the various types trust models and cloud security.

UNIT-I: INTRODUCTION: SECURING THE INTERNET OF THINGS 9

Security Requirements in IoT Architecture – Security in Enabling Technologies – Security Concerns in IoT Applications. Security Architecture in the Internet of Things – Security Requirements in IoT – Insufficient Authentication/Authorization – Insecure Access Control – Threats to Access Control, Privacy and Availability – Attacks Specific to IoT. Vulnerabilities – Secrecy and Secret – Key Capacity – Authentication/Authorization for Smart Devices – Transport Encryption – Attack & Fault trees.

UNIT-II: CRYPTOGRAPHIC FUNDAMENTALS FOR IOT 9

Cryptographic primitives and its role in IoT – Encryption and Decryption – Hashes – Digital signatures – Random number generation – Cipher suites – Key management fundamentals – Cryptographic controls built into IoT messaging and communication protocols – IoT Node Authentication.

UNIT-III: IDENTITY & ACCESS MANAGEMENT SOLUTIONS FOR IOT 9

Identity lifecycle – Authentication credentials – IoT IAM infrastructure – Authorization with Publish / Subscribe schemes – Access control.

UNIT-IV: PRIVACY PRESERVATION AND TRUST MODELS FOR IOT 9

Concerns in data dissemination – Lightweight and robust schemes for privacy protection – Trust and Trust models for IoT – Self – organizing things – Preventing unauthorized access.

UNIT-V: CLOUD SECURITY FOR IOT 9

Cloud services and IoT – Offerings related to IoT from cloud service providers – Cloud IoT security controls – An enterprise IoT cloud security architecture – New directions in cloud enabled IoT computing.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Brian Russell and Drew Van Duren, "Practical Internet of Things Security (Kindle Edition) by Brian Russell, Drew Van Duren" , Packt Publishing, 2016.
2. Securing the Internet of Things Elsevier.
3. Fei HU. "Security and Privacy in Internet of Things (IoTs): Models, Algorithms and Implementations", CRC Press.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: The security requirements in IoT.

CO2: The cryptographic fundamentals for IoT.

CO3: The authentication credentials and access control.

CO4: Various types trust models and cloud security.

19CSPE104	EMBEDDED SYSTEMS	SEMESTER I			
		L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- To understand the architecture and instruction set of microcontrollers.
- Ability to understand addressing modes and interrupt mechanisms of microcontrollers.
- To understand peripheral functions, timers and data convertors and their interfacing.
- To understand RTOS, multiple process environment and develop applications.
- To use development tools and hardware software co - design.

UNIT-I: MICROCONTROLLER ARCHITECTURE, CLOCK AND OPERATING MODES **9**

New generation embedded systems: Low power operations, High performance, Battery operated embedded systems; Introduction to RL78 microcontrollers; Architecture of RL78 microcontrollers, General purpose registers; Memory space; Flash mirror facility; Boot clusters; Special function registers; Pipeline execution. RL78 clock circuitry and operating modes; Operating modes; Reset management; Power – on – reset; Voltage detection circuit; Applying voltage detection circuits.

UNIT-II: INSTRUCTION SET AND FAIL: SAFE FEATURES **9**

Instruction set; Addressing modes; Types of instructions; Types of interrupts; Interrupt sources and configurations, Interrupt priority; Interrupt servicing; Key interrupt functions; Introduction to fail– safe standard IEC60730; Usage of CRC in memory; Detection of abnormal CPU operations.

UNIT-III: PERIPHERALS: I/O PORTS, COMMUNICATION FUNCTIONS, TIMERS, DATA CONVERTERS **9**

RL78 peripheral functions; I/O Ports; Port architecture; Port operations; Port controlling registers; Serial ports of RL78, Functions of 3-wire serial I/O; Functions of UART channels; Functions of simplified IIC channels; Functions of LIN communications, Timer array units; PWM output generation; One-shot pulse outputs; Multiple PWM outputs; Interval timers; Real time counters; Watchdog timers; Analog to digital converter overview; A/D conversion operations; A/D conversion modes; Flash memory configurations; Flash memory programming.

UNIT-IV: INTRODUCTION TO ARM CORTEX M3 MICROCONTROLLERS **9**

Introduction to STM32F1xx family, Overview of Cortex – M3 architecture, Bus configurations and Memory structure, Reset and Clock circuitry, General purpose and alternate function I/Os, Interrupts and events, DMA controller, Data converters, Timers, Watchdog timers, Flexible static memory controller, SDIO, communication facilities like SPI, IIC, CAN, Ethernet, USB.

UNIT-V: RTOS, DEVELOPMENT TOOLS AND HARDWARE SOFTWARE CO-DESIGN **9**

Understanding code development environment for microcontrollers, Debugging tools, Embedded system design methodologies, RTOS, Hardware software code design.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Alexander G. Dean and James M. Conard, " Creating Fast, Responsive and Energy – efficient Embedded Systems using the Renesas RL 78 Microcontroller", Micrium Press, 2011.
2. Joseph Yiu, "The Definitive Guide to the ARM Cortex-M3", Elsevir Inc., 2nd Edition, 2010.
3. Frank Vahid and Tony D.Givargis, "Embedded System Design: A Unified Hardware/Software Introduction", John Wily & Sons Inc.2002.
4. Peter Marwedel, "Embedded System Design", Science Publishers, 2007.
5. Tammy Noergaard, "Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers", Elsevier Pvt. Ltd. Publications, 2005.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Describe architectural features of RENESAS RL78 microcontroller.

CO2: Describe the multiple process operating environment and system call interfaces to monitor and control processes.

CO3: Develop interface peripherals for serial communication, timer applications and Data convertors.

CO4: Describe architectural features of ARM Cortex M3 Microcontroller.

CO5: Design and implement software systems to provide an interface to ARM Cortex M3 based hardware systems.

19CSPE105 ADVANCED COMPUTER ARCHITECTURE SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To introduce the students to the recent trends in the field of Computer Architecture and identify performance related parameters.
- To learn the different multiprocessor issues.
- To expose the different types of multicore architectures.
- To understand the design of the memory hierarchy.

UNIT-I: FUNDAMENTALS OF COMPUTER DESIGN AND ILP 9

Fundamentals of computer design – Measuring and reporting performance – Instruction level Parallelism and its exploitation – Concepts and challenges – Exposing ILP – Advanced branch Prediction – Dynamic scheduling – Hardware – Based speculation – Exploiting ILP – Instruction delivery and speculation – Limitations of ILP – Multithreading.

UNIT-II: MEMORY HIERARCHY DESIGN 9

Introduction – Optimizations of cache performance – Memory technology and optimizations – Protection: Virtual memory and virtual machines – Design of memory Hierarchies – Case studies.

UNIT-III: MULTIPROCESSOR ISSUES 9

Introduction – Centralized, Symmetric and distributed shared memory architectures – Cache coherence Issues – Performance issues – Synchronization – Models of memory consistency – Case study – Interconnection Networks – Buses, Crossbar and multi – Stage interconnection networks.

UNIT-IV: MULTICORE ARCHITECTURES 9

Homogeneous and heterogeneous multi – Core architectures – Intel multicore architectures – SUN CMP architecture – IBM cell architecture. Introduction to warehouse – Scale computers Architectures – Physical infrastructure and Costs – Cloud computing – Case study – Google warehouse – Scale computer.

UNIT-V: VECTOR, SIMD AND GPU ARCHITECTURES 9

Understanding code development environment for microcontrollers, Debugging tools – Embedded system design methodologies – RTOS – Hardware software code design.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Darryl Gove, “Multicore Application Programming: For Windows, Linux, and Oracle Solaris”, Pearson, 2011.
2. David B. Kirk, Wen - Mei W. Hwu, “Programming Massively Parallel Processors”, Morgan Kauffman, 2010.
3. David E. Culler and Jaswinder Pal Singh, “Parallel computing architecture : A hardware/software approach”, Morgan Kaufmann /Elsevier Publishers, 1999.

4. John L. Hennessey and David A. Patterson, “Computer Architecture – A Quantitative Approach”, Morgan Kaufmann / Elsevier, 5^{th} Edition, 2012.
5. Kai Hwang and Zhi.Wei Xu, “Scalable Parallel Computing”, Tata McGraw Hill, NewDelhi, 2003.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Identify the limitations of ILP.

CO2: Discuss the issues related to multiprocessing and suggest solutions.

CO3: Point out the salient features of different multicore architectures and how they exploit parallelism.

CO4: Discuss the various techniques used for optimizing the cache performance.

CO5: Design hierarchal memory system. Point out how data level parallelism is exploited in architectures.

19CSPE106	WEB ENGINEERING	SEMESTER I
		L T P C
		3 0 0 3

COURSE OBJECTIVES:

- Acquire knowledge in hypertext markup language and cascading style sheet.
- Understand client side programming using java script.
- Understand server side programming using servlet, JSP and PHP.
- Acquire knowledge in XML and Ajax enabled Internet application design.
- Understand server side development.

UNIT-I: INTRODUCTION TO WEBSITES , HTML5 AND CSS3 9

Introduction to Internet, Websites and Web Servers, Internet and Intranet, Web 1.0 vs Web 2.0 vs Web 3.0, HTML 5: Basic HTML Elements, Input and page structure elements, Positioning elements, Backgrounds, Element dimensions, Box model and text flow, Media types and queries, Shadows, Gradients, Animations, Transitions and transformations, Web font, Multi column layout – Cascading style Sheet 3: Inline, Internal and external CSS.

UNIT-II: CLIENT SIDE SCRIPTING 9

Java script: Programming basics, Introduction to scripting, Control statement, Functions, objects, Event handling, Regular expressions, Exception handling, Validation – Built in objects: Math, String, Date, Arrays, Boolean, Document objects – Document object model.

UNIT-III: SERVER SIDE PROGRAMMING 9

Servlets: Java Servlet Architecture, Servlet life cycle – Form GET and POST actions, Session handling, Understanding cookies, Installing and configuring Apache Tomcat Web server Database Connectivity: JDBC perspectives, JDBC program example, JSP: Understanding java server pages-JSP Standard Tag Library(JSTL), Creating HTML forms by embedding JSP code, An introduction to PHP: PHP – Using PHP, Variables, Program control – Built – in functions – Connecting to database, Using cookies – Regular expressions.

UNIT-IV: XML, JSON and AJAX ENABLED RICH INTERNET APPLICATIONS 9

XML: Basics, Structuring data, XML name spaces, DTDs – Schema documents, Extensible style sheet Language and XSL transformation, DOM, Web application development: Traditional Vs Ajax web application development, RIA with Ajax, XML Http request object, Using XML and DOM, Application creation.

UNIT-V: SERVER– SIDE DEVELOPMENT WITH JSF AND JAVA 9

Java Server Faces: Application development, Model view controller architecture, JSF Components, Validation, Session Tracking, Accessing databases in Web Apps, Web Services: SOAP, REST, JSON, Publishing and consuming SOAP based web services, REST based XML Web services, REST based JSON Web service.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Paul Deitel, Harvey Deitel, Abbey Deitel "Internet and World Wide Web - How to Program"Fifth Edition, Pearson,2012.
2. Achyut Godbole, Atul Kahate, "Web Technologies:TCP/IP to Internet Application Architectures", Tata McGraw-Hill Education, 2002
3. Jon Duckett, "Beginning Web Programming with HTML, XHTML and CSS", Wrox Press,2004.
4. C. Zakas, "Professional Javascript for Web Developers", Third Edition, Wrox Press,2011.
5. Gopalan N.P. and Akilandeswari J., "Web Technology", Prentice Hall of India, 2011.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Develop Websites using Hypertext Markup language and Cascading Style Sheet.

CO2: Design Client Side Programming using JavaScript.

CO3: Write Server Side Programming using Servlet, JSP and PHP.

CO4: Acquire Knowledge in XML and Ajax enabled Internet application design.

CO5: Create Server Side Application using JSF and Java.

19CSPE107 DEEP LEARNING SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- Understand fundamentals of Machine learning.
- Acquire knowledge in deep feed forward networks.
- Understand convolution Neural network.
- Acquire knowledge auto Encoders.

UNIT-I: FUNDAMENTALS CONCEPTS OF MACHINE LEARNING 9

historical trends in deep learning – Machine learning basics: learning algorithms – Supervised and unsupervised training, Linear algebra for machine learning, Testing, cross – Validation, dimensionality reduction, over/under-fitting, Hyper parameters and validation sets, Estimators, bias, Variance, regularization – Introduction to a simple dnn, Platform for deep learning, Deep learning software libraries.

UNIT-II: DEEP FEED FORWARD NETWORKS 9

Deep feed forward networks – Introduction – Learning XOR – Gradient-based learning- Various activation functions, Error functions – Architecture design – Differentiation algorithms regularization for deep learning – Early stopping, Drop out.

UNIT-III: CONVOLUTIONAL NEURAL NETWORKS AND SEQUENCE MODELING 9

Convolutional networks: Convolutional operation – Motivation – Pooling – Normalization, Applications in computer vision: Imagenet sequence modeling: Recurrent neural networks difficulty in training RNN – Encoder – Decoder

UNIT-IV: AUTO ENCODERS 9

Auto encoders – Auto encoders: Under complete, regularized, Stochastic, Denoising, contractive, applications – Dimensionality reduction, Classification, Recommendation, Optimization for deep learning: Optimizers. RMS prop for RNNs, SGD for CNNs.

UNIT-V: DEEP ARCHITECTURES IN VISION 9

Deep Architectures in vision – Alexnet to ResNet, Transfer learning, Siamese networks, Metric Learning, Ranking/Triplet loss, RCNNs, CNN-RNN, Applications in captioning and video tasks, 3D CNNs.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, “Deep Learning”, MIT Press, 2016.
2. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
3. Li Deng and Dong Yu, “Deep Learning: Methods and Applications”, Foundations and Trends in Signal Processing.
4. Christopher and Bishop M., “Pattern Recognition and Machine Learning”, Springer Science Business Media, 2006.

5. Lewis N.D., "Deep Learning Step by Step with Python: A Very Gentle Introduction to Deep Neural Networks for Practical Data Science.
6. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.
7. Christopher Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Distinguish between, supervised, unsupervised and semi-supervised learning.
CO2: Apply the deep feed forward network.
CO3: Use convolution Neural network.
CO4: Apply auto encoders.
CO5: Design systems that uses the deep Architecture.

19CSPE108 WIRELESS SENSOR NETWORKS SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand Wireless Sensor Networks.
- To illustrate Architecture of Wireless Sensor Networks.
- To use Sensors for networking.
- To Establish the Infrastructure for WSN.
- To use Sensor Network platforms and tools.

UNIT- I: INTRODUCTION 9

Challenges for Wireless Sensor Networks, Characteristics requirements, Required mechanisms, Difference between mobile ad-hoc and sensor networks, Applications of sensor networks, Enabling Technologies for Wireless Sensor Networks.

UNIT- II: ARCHITECTURES 9

Single node architecture, Hardware components, Energy consumption of sensor nodes, Operating systems and execution environments, Network architecture, Sensor network scenarios, Optimization goals and figures of merit, Gateway concepts.

UNIT-III: NETWORKING OF SENSORS 9

Physical layer and transceiver design considerations, Mac protocols for wireless sensor networks, Low duty cycle protocols and wakeup concepts, S-mac, the mediation device protocol, Wakeup radio concepts, Address and name management, Assignment of mac addresses, Routing protocols, energy – Efficient routing, geographic routing.

UNIT-IV: INFRASTRUCTURE ESTABLISHMENT 9

Topology control, Clustering, Time synchronization, Localization and positioning, Sensor tasking and control.

UNIT-V: SENSOR NETWORK PLATFORMS AND TOOLS 9

Operating systems for wireless sensor networks, sensor node hardware – berkeley motes, programming challenges, node – Level software platforms, Node – level Simulators, State–centric programming.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks”, John Wiley, 2005.
2. Feng Zhao and Leonidas J. Guibas, “Wireless Sensor Networks-An Information Processing Approach”, Elsevier, 2007.
3. Kazem Sohraby, Daniel Ivlinoli, and Taieb Znati, “Wireless Sensor Networks Technology, Protocols and Applications”, John Wiley, 2007.
4. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.
5. Bhaskar Krishnamachari, “Networking Wireless Sensors”, Cambridge Press, 2005.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the characteristics, requirements and applications of Wireless Sensor Networks.

CO2: Explain Architecture of Wireless Sensor Networks.

CO3: Illustrate MAC and routing protocols of WSN.

CO4: Establish infrastructure for WSN.

CO5: Use Sensor Network platforms and tools.

19CSPE109 INTRODUCTION TO INTELLIGENT SYSTEMS SEMESTER I

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To Introduce to the field of Artificial Intelligence (AI) with emphasis on its use to solve real world problems for which solutions are difficult to express using the traditional algorithmic approach.
- To explore the essential theory behind methodologies for developing systems that demonstrate intelligent behavior including dealing with uncertainty, learning from experience and following problem solving strategies found in nature.

UNIT-I: BIOLOGICAL FOUNDATIONS TO INTELLIGENT SYSTEMS I 9

Artificial neural networks, Back propagation networks, Radial basis function networks, and recurrent networks.

UNIT-II: BIOLOGICAL FOUNDATIONS TO INTELLIGENT SYSTEMS II 9

Fuzzy logic, knowledge Representation and inference mechanism, genetic algorithm and Fuzzy neural networks.

UNIT-III: SEARCH METHODS 9

Basic concepts of graph and tree search. Three simple search methods: breadth-first search, depth-first search, iterative deepening search. Heuristic search methods: best-first search, admissible evaluation functions, Hill-Climbing search. Optimization and search such as stochastic annealing and genetic algorithm.

UNIT-IV: KNOWLEDGE REPRESENTATION 9

Knowledge representation and logical inference Issues in knowledge representation. Structured representation, such as frames, and scripts, semantic networks and conceptual graphs. Formal logic and logical inference. Knowledge based systems structures, its basic components. Ideas of blackboard architectures.

UNIT-V: REASONING 9

Reasoning under uncertainty and learning techniques on uncertainty reasoning such as bayesian reasoning, certainty factors and dempster – Shafer theory of evidential reasoning, a study of different learning and evolutionary algorithms, such as statistical learning and induction learning. Recent trends in Fuzzy logic, Knowledge representation.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Luger G. F and Stubblefield W.A., "Artificial Intelligence: Structures and strategies for Complex Problem Solving", Addison Wesley, 6th Edition, 2008.
2. Russell S and Norvig P., "Artificial Intelligence: A Modern Approach. Prentice – Hall", 3rd Edition, 2009.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Demonstrate knowledge of the fundamental principles of intelligent systems.

CO2: Analyze and compare the relative merits of a variety of AI problem solving techniques.
CO3: Illustrate search methods.
CO4: Establish knowledge-based system structures.
CO5: Use uncertainty and learning techniques.

PROFESSIONAL ELECTIVES – II

19CSPE201 **DATA SCIENCE** **SEMESTER II**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To learn about the basics of data Science and to understand the various supervised and unsupervised learning techniques.
- To bring together several key technologies used for manipulating, storing, and analyzing big data from advanced analytics perspectives.
- To realize the hadoop architecture and implementation of map reduce application.

UNIT-I: INTRODUCTION TO DATA SCIENCE **9**

Introduction of data science, Basic data analytics using R, R graphical user interfaces. Data import and export, Attribute and data types, Descriptive statistics, Exploratory data analysis, Visualization before analysis, Dirty data, Visualizing a single variable, Examining multiple variables, Data exploration versus presentation. Statistical methods for evaluation, Hypothesis testing, Difference of means, Wilcoxon rank – Sum test, ANOVA.

UNIT-II: ADVANCED ANALYTICAL THEORY AND METHODS **9**

Overview of clustering, K- means, Use cases, Overview of the method, Perform a K- means analysis using R. Classification, Decision trees, Overview of a decision tree, Decision tree algorithms, Evaluating a decision tree. Decision tree in R, Bayes' theorem, Naive Bayes Classifier, Smoothing, Naive Bayes in R.

UNIT-III: ADVANCED ANALYTICS TECHNOLOGY AND TOOLS **9**

Analytics for unstructured data, Use cases, Map reduce, Apache Hadoop, The hadoop Ecosystem, Pig, hive, Hbase, Mahouth, NoSQL, SQL essentials. Joins, Set operations, Grouping extensions, In-Database Text analysis, Advanced SQL, Window functions, User – defined functions and aggregates, Ordered aggregates, MAD lib.

UNIT-IV: HADOOP DISTRIBUTED FILE SYSTEM ARCHITECTURE AND SPARK **9**

HDFS Architecture, HDFS concepts, Blocks. Name node, Secondary Name node, Data node, HDFS Federation, HDFS high availability, Basic file system operations. Data flow, Anatomy of file read, Anatomy of file write, Anatomy of a map reduce job run. spark – Features of spark – Spark built on Hadoop – Components of Spark – Mllib and machine learning.

UNIT-V: PROCESSING DATA WITH MAPREDUCE **9**

Getting to know map reduce, Map reduce execution pipeline, Runtime coordination and task management, Map reduce application, Hadoop word count implementation. Installing and running pig, Hbase versus RDBMS, Installing and running zoo keeper.

Contact Periods:

Lecture: 45 Periods **Tutorial: 0 Periods** **Practical: 0 Periods** **Total: 45 Periods**

REFERENCES

1. David Dietrich, Barry Heller and Beibei Yang, “Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data”, EMC Education Services, Reprint 2015.
2. Tom White, "Hadoop: The Definitive Guide", O'Reilly, 4th Edition, 2015.
3. Biris Lublinsky, Kevin T. Smith and Alexey Yakubovich, “Professional Hadoop Solutions”, Wiley, Reprint 2014.
4. Stephen Marsland, “Machine Learning – An Algorithmic Perspective”, Taylor& Francis Group, Chapman & Hall / CRC Press, 2nd Edition, 2015.
5. Nathan Marz, James Warren, “Big Data – Principles and Best Practices of Scalable Real–Time Data Systems”, Dream Tech Press, Edition 2015.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Acquire knowledge about the basics of data Science and various supervised and unsupervised learning techniques.

CO2: Explain analytical theory and methods.

CO3: nonstrate advanced analytical technology and tools.

CO4: 'elop programs in Hadoop using HDFS and spark.

CO5: Process the data with Map reduce.

19CSPE202 ADVANCED DISTRIBUTED SYSTEMS SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To learn of the concepts, principles and technologies of distributed systems.
- To introduce advanced idea of peer to peer and file system management.
- To understand the issues involved in resource management and process.

UNIT-I: DISTRIBUTED SYSTEMS 9

Introduction to distributed systems – Characterization of distributed systems – Distributed architectural models – Remote invocation – Request– Reply protocols – RPC – RMI – Group communication – Coordination in group communication – Ordered multicast – Time ordering – Physical clock synchronization – Logical time and logical clocks.

UNIT-II: DISTRIBUTED SECRUITY AND TRANSACTIONS 9

Introduction – Overview of security techniques – Cryptographic algorithms – Digital signatures – Cryptography pragmatics – Flat and nested distributed transactions – Atomic commit protocols – Concurrency control in distributed transactions – Distributed deadlocks – Transaction recovery.

UNIT-III: DISTRIBUTED MUTUAL EXCLUSION ALGORITHMS 9

Introduction – Lamport's algorithm – Ricart Agrawala algorithms – Singhal's dynamic information structure algorithm – Lodha and Kshemkalyani's fair mutual exclusion algorithms – Quorum based algorithm – Mackawa's algorithms – Token based algorithms – Roymaond's tree based algorithms.

UNIT-IV: DEADLOCK DETECTION IN DISTRIBUTION SYSTEMS 9

System Model – Models of deadlocks – Knapp's classification of distributed deadlock detection algorithms – Mitchell & Merritt's algorithm for the single resource model – Chandy Misra Haas algorithm for the AND & OR Model – Kshemkalyanisinghal algorithm for P out of Q model – Global predicate detection.

UNIT-V: ADVANCED IN DISTRIBUTED SYSTEMS 9

Authentication in distributed systems – Protocols based on symmetric cryptosystems – Protocols based on asymmetric cryptosystems – Password-based authentication – Authentication protocol failures – Self-Stabilization – Peer-to-peer computing and overlay graphs – Unstructured overlays – Chord distributed hash table – Content addressable networks (CAN) – Tapestry – Some other challenges in P2P system design – Tradeoffs between table storage and route lengths – Graph structures of complex networks – Internet graphs – Generalized random graph networks – Small world networks – Scale-free networks – Evolving networks.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education Asia, 2012.

2. Ajay D. Kshemkalyani, MukeshSinghal, “Distributed Computing: Principles, Algorithms, and Systems”, Cambridge University Press, 2008.
3. Liu, “Distributed Computing: Principles and Applications”, Pearson Education , 2004.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Design principles and technologies of Distributed systems.

CO2: Know about advanced idea of peer to peer and file system management.

CO3: Identify the issues involved in resource management and process.

CO4: Develop programs in Deadlock detection.

CO5: Point out the salient features of Advanced in distributed systems.

19CSPE203 ARTIFICIAL NEURAL NETWORKS SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To know the fundamental theory and concepts of neural networks.
- To be familiar with essentials of Artificial Neural Networks.
- To understand single layer feed forward Networks.
- To understand Multi - Layer feed forward Networks and Associative memories.

UNIT-I: INTRODUCTION TO NEURAL NETWORKS 9

Introduction, Humans and computers, Organization of the brain, Biological Neuron, Biological and Artificial Neuron models, Characteristics of ANN, McCulloch– Pitts model, Historical developments, Potential applications of ANN.

UNIT-II: ESSENTIALS OF ARTIFICIAL NEURAL NETWORKS 9

Artificial Neuron model, Operations of Artificial Neuron, Types of Neuron Activation function, ANN Architectures, Classification Taxonomy of ANN – Connectivity, Learning strategy (Supervised, Unsupervised, Reinforcement), Learning rules.

UNIT-III: SINGLE LAYER FEED FORWARD NETWORKS 9

Introduction, Perceptron models: Discrete, Continuous and multi – Category, Training Algorithms: Discrete and continuous perceptron Networks, Limitations of the perceptron model.

UNIT-IV: MULTI-LAYER FEED FORWARD NETWORKS 9

Credit assignment problem, Generalized delta rule, Derivation of Back Propagation (BP) Training, Summary of Back propagation Algorithm, Kolmogorov theorem, Learning difficulties and improvements.

UNIT-V: ASSOCIATIVE MEMORIES 9

Paradigms of associative memory, Pattern Mathematics, Hebbian learning, General concepts of associative memory, Bidirectional Associative Memory (BAM) Architecture, BAM training Algorithms: Storage and recall algorithm, BAM energy function. Architecture of hopfield Network: Discrete and continuous versions, Storage and recall algorithm, Stability analysis. Neural network applications: Process identification, Control, Fault diagnosis.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Laurene Fausett, “Fundamentals of Neural Networks” , Pearson Education, 2004.
2. Simon Haykin, “Neural Networks – A comprehensive foundation”, Pearson Education, 2003.
3. Sivanandam S N , Sumathi S, Deepa S.N., “Introduction to Neural Networks using MATLAB 6.0”, TATA Mc Graw Hill, 2006.
4. Rajasekharan S and Vijayalakshmi Pai S.A., “Neural Networks, Fuzzy logic, Genetic Algorithms: Synthesis and Applications”, PHI Publication, 2004.
5. Timothy J. Ross, “Fuzzy Logic With Engineering Applications”, Tata McGraw – Hill Inc. 2000.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the Fundamental theory and concepts of neural networks.

CO2: Design simple Artificial Neural Network.

CO3: 'elop single layer feed forward Networks.

CO4: ognize multi - layer feed forward Networks.

CO5: iew the concepts of associative memories.

19CSPE204 THEORY OF MODERN COMPILERS SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand intermediate representations.
- To be familiar with control and data flow analysis.
- To know early and loop optimization.
- To understand procedure optimization and scheduling.
- To perform inter procedural analysis and memory hierarchy optimization.

UNIT-I: INTERMEDIATE REPRESENTATIONS 9

Introduction to compiler technologies, Review of compiler structure, Intermediate representations – Run time support: Data representations and instructions, Register usage, the local stack frame, run time stack, Parameter passing, Procedure prologues, Epilogues, Call and returns, Code sharing and position independent Code – Producing code generators automatically.

UNIT-II: FLOW ANALYSIS 9

Control flow analysis, Data flow analysis: Iterative data flow analysis, Lattices of flow functions, Control tree based data flow analysis, Structural analysis, Interval analysis, Dependence analysis and dependence graph – Alias analysis.

UNIT-III: EARLY OPTIMIZATIONS AND LOOP OPTIMIZATIONS 9

Introduction to optimization, Importance of individual optimizations, Order and repetition of optimizations, Early optimization: Constant folding, Scalar replacement of aggregates, Algebraic simplifications and reassociation, Value numbering, Copy and constant propagation, Redundancy elimination, Loop optimizations.

UNIT-IV: PROCEDURE OPTIMIZATION AND SCHEDULING 9

Procedure optimizations– Register allocation, Code scheduling, Control – Flow and low– level optimizations: Unreachable code elimination, Straightening, If and loop simplification, Loop inversion, Un switching, Branch optimizations, Tail merging, Conditional moves, Dead code elimination, Branch prediction.

UNIT-V: INTERPROCEDURAL ANALYSIS AND MEMORY HIERARCHY OPTIMIZATION 9

Inter procedural analysis and optimizations: Control flow, Data flow and Alias analysis, Constant propagation, Optimization and register allocation, Optimization for the memory hierarchy: Impact of data and instruction caches and optimizations.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Aho V, Monical Lam, Sethi R, Ullman J D.,"Compilers: Principles, Techniques and Tools", 2^{nd} Edition , 2008.
2. Steven Muchnick.,"Advanced Compiler Design and Implementation", Morgan Kaufmann Publishers, Elsevier, 2008.

3. Randy Allen and Ken Kennedy, “Optimizing Compilers for Modern Architectures”, Morgan Kaufmann, Elsevier, 2002.
4. Andrew W. Appel, Jens Palsberg, “Modern Compiler Implementation in Java”, 2nd Edition, Cambridge University Press, 2002.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Outline Intermediate representations.
CO2: Perform control and data flow analysis.
CO3: ninate redundancy from IR and target code.
CO4: imize loops, Procedures and memory hierarchy.
CO5: ıerate target code.

19CSPE205 MOBILE AND PERVASIVE COMPUTING SEMESTER II

	L	T	P	C
	3	**0**	**0**	**3**

COURSE OBJECTIVES:

- To learn the basic architecture and concepts till third generation communication systems.
- To understand the latest 4G telecommunication system principles.
- To introduce the broad perspective of pervasive concepts and management.
- To explore the HCI in pervasive environment.
- To apply the pervasive concepts in mobile environment.

UNIT-I: INTRODUCTION 9

History – Wireless communications: GSM – DECT – TETRA – UMTS – IMT – 2000 – Blue tooth, WiFi, WiMAX, 3G, WATM.– Mobile IP protocols – WAP push architecture – Wml scripts and applications. Data networks – SMS – GPRS – EDGE – Hybrid Wireless100 Networks – ATM – Wireless ATM.

UNIT-II: OVERVIEW OF A MODERN 4G TELECOMMUNICATIONS SYSTEM 9

Introduction. LTE – A system architecture. LTE RAN. OFDM Air interface. Evolved packet Core. LTE requirements. LTE – Advanced. LTE – A in release. OFDMA – Introduction. OFDM principles. LTE Uplink – SC – FDMA. Summary of OFDMA.

UNIT-III: PERVASIVE CONCEPTS AND ELEMENTS 9

Technology trend overview – Pervasive computing: concepts – Challenges – Middleware – Context awareness – Resource management – Human– Computer interaction – Pervasive transaction processing – Infrastructure and devices – Wireless Networks – Middleware for pervasive computing systems – Resource management – User tracking – Context management – Service management – Data management – Security management – Pervasive computing environments – Smart car space – Intelligent campus.

UNIT-IV: HCI IN PERVASIVE COMPUTING 9

Prototype for application migration – Prototype for multimodalities – Human– Computer interface in pervasive environments – HCI service and interaction migration – Context – Driven HCI service selection – Interaction service selection overview – User devices – Service – Oriented middleware support – User history and preference – Context manager – Local service matching – Global combination – Effective region – User active scope – Service combination selection algorithm.

UNIT-V: PERVASIVE MOBILE TRANSACTIONS 9

Pervasive mobile transactions – Introduction to pervasive transactions – Mobile transaction framework – Unavailable transaction service – Pervasive transaction processing framework – Context– Aware pervasive transaction model – Context model for pervasive transaction processing – Context – Aware pervasive transaction model – A case of pervasive transactions – Dynamic transaction management – Context – Aware transaction coordination mechanism – Coordination algorithm for pervasive transactions – Participant discovery – Formal transaction verification – Petri net with selective transition.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Alan Colman, Jun Han and Muhammad Ashad Kabir, “Pervasive Social Computing Socially– Aware Pervasive Systems and Mobile Applications”, Springer, 2016.
2. Schiller J., “Mobile Communication”, Addison Wesley, 2000.
3. Juha Korhonen, “Introduction to 4G Mobile Communications”, Artech House Publishers, 2014.
4. Kolomvatsos and Kostas, “Intelligent Technologies and Techniques for Pervasive Computing”, IGI Global, 2013.
5. Bala Krishna M, Jaime Lloret Mauri, “Advances in Mobile Computing and Communications: Perspectives and Emerging Trends in 5G Networks”, CRC 2016.
6. Minyi Guo, Jingyu Zhou, Feilong Tang, Yao Shen, “Pervasive Computing: Concepts, Technologies and Applications”, CRC Press, 2016.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Obtain a thorough knowledge of Basic architecture and concepts of till Third Generation Communication systems.

CO2: Explain the latest 4G Telecommunication System Principles.

CO3: orporate the pervasive concepts.

CO4: lement the HCI in Pervasive environment.

CO5: rk on the pervasive concepts in mobile environment.

19CSPE206 PARALLEL PROGRAMMING PARADIGMS SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To familiarize the issues in parallel computing.
- To describe distributed memory programming using MPI.
- To understand shared memory paradigm with Pthreads.
- To develop shared– memory parallel programs using OpenMP
- To learn the GPU based parallel programming using OpenCL.

UNIT-I: FOUNDATIONS OF PARALLEL PROGRAMMING 9

Motivation for parallel programming – Need – Concurrency in computing – Basics of processes, Multitasking and threads – Cache – Cache mappings – Caches and programs – Virtual memory – Instruction level parallelism – Hardware multi– Threading – Parallel Hardware– SIMD – MIMD – Interconnection networks – Cache coherence – Issues in shared memory model and distributed memory model – Parallel Software – Caveats – Coordinating processes/ threads – Hybrid model – Shared memory model and distributed memory model – I/O – Performance of parallel programs – Parallel program design.

UNIT-II: DISTRIBUTED MEMORY PROGRAMMING WITH MPI 9

Basic MPI programming – MPI_Init and MPI_Finalize – MPI communicators – SPMD programs– MPI_Send and MPI_Recv – Message matching – MPI– I/O – Parallel I/O – collective communication – Tree – Structured communication – MPI_Reduce – MPI_Allreduce, Broadcast, Scatter, gather, Allgather – MPI derived types – Dynamic process management – Performance evaluation of MPI programs – A Parallel Sorting Algorithm.

UNIT-III: SHARED MEMORY PARADIGM WITH PTHREADS 9

Basics of threads, Pthreads – Thread synchronization – Critical sections – Busy waiting – mutex – Semaphores – Barriers and condition variables – Read write locks with examples – Caches, Cache coherence and false sharing – Thread safety – Pthreads case study.

UNIT-IV: SHARED MEMORY PARADIGM: OPENMP 9

Basics open MP – Trapezoidal rule – Scope of variables – Reduction clause – Parallel for directive – Loops in OpenMP – Scheduling loops – Producer Consumer problem – Cache issues – Threads safety in Open MP – Two – body solvers –Tree Search.

UNIT-V: GRAPHICAL PROCESSING PARADIGMS: OPENCL AND INTRODUCTION TO CUDA 9

Introduction to open CL – Example – Open CL platforms – Devices – Contexts – Open CL programming – Built– In Functions – Programs object and Kernel object – Memory objects – Buffers and images – Event model – Command– Queue – Event object – Case study. Introduction to CUDA programming.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Munshi A, Gaster B, Mattson. T G, Fung J and Ginsburg D., "Open CL Programming Guide", Addison Wesley, 2011.
2. Quinn M J., "Parallel Programming in C with MPI and Open MP", Tata McGraw Hill, 2003.
3. Peter S. Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann, 2011.
4. Rob Farber, "CUDA Application Design and Development", Morgan Haufmann, 2011.
5. Gropp W, Lusk E and Skjellum A., "Using MPI: Portable Parallel Programming with the Message Passing Interface", 2nd Edition, MIT Press, 1999.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Identify issues in parallel programming.

CO2: Develop distributed memory programs using MPI framework.

CO3: Design and implement shared memory parallel programs using Pthreads.

CO4: Extend and develop shared memory parallel programs using Open MP.

CO5: Implement Graphical Processing Open CL programs.

19CSPE207 INFORMATION STORAGE MANAGEMENT SEMESTER II

	L	T	P	C
	3	**0**	**0**	**3**

COURSE OBJECTIVES:

- To familiarize the data model.
- To understand data security and privacy.
- To understand information governance.
- To understand information architecture.
- To learn the information life cycle management.

UNIT-I: DATABASE MODELLING, MANAGEMENT AND DEVELOPMENT 9

Database design and modelling – Business rules and relationship; Java database Connectivity (JDBC), Database connection manager, Stored procedures. Trends in big data systems including NoSQL – Hadoop HDFS, Map reduce, Hive and Enhancements.

UNIT-II: DATA SECURITY AND PRIVACY 9

Program Security, Malicious code and controls against threats; OS level protection; Security – Firewalls, Network security intrusion detection systems. Data privacy principles. Data privacy laws and compliance.

UNIT-III: INFORMATION GOVERNANCE 9

Master data management (MDM) – Overview, Need for MDM, Privacy, Regulatory requirements and compliance. Data governance – Synchronization and data quality management.

UNIT-IV: INFORMATION ARCHITECTURE 9

Principles of information architecture and framework, Organizing information, Navigation systems and Labelling systems, Conceptual design, Granularity of content.

UNIT-V: INFORMATION LIFECYCLE MANAGEMENT 9

Data retention policies; Confidential and sensitive data handling, lifecycle management costs. Archive data using Hadoop; Testing and delivering big data applications for performance and functionality; Challenges with data administration.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Alex Berson, Larry Dubov, "Master Data Management and Data Governance", 2nd Edition, Tata McGraw Hill, 2011.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in Computing", 4th Edition, Prentice Hall; 2006.
3. Peter Morville, Louis Rosenfeld, "Information Architecture for the World Wide Web", O'Reilly Media; 1998.
4. Jeffrey A. Hoffer, Heikki Topi, Ramesh V., "Modern Database Management", 10th Edition, Pearson, 2012.

5. Jeffrey Carr, “Inside Cyber Warfare: Mapping the Cyber Underworld”, O'Reilly Media; 2nd Edition 2011.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Develop database model.

CO2: Know data security and privacy.

CO3: Design and develop information architecture.

CO4: Interpret information life cycle management.

19CSPE208	**ADVANCED WIRELESS AND MOBILE NETWORKS**	**SEMESTER II**
		L T P C
		3 0 0 3

COURSE OBJECTIVES:

- The students should get familiar with the wireless/mobile market and the future needs and challenges.
- To get familiar with key concepts of wireless networks, standards, technologies and their basic operations.
- To learn how to design and analyze various medium access.
- To learn how to evaluate MAC and network protocols using network simulation software tools.
- The students should get familiar with the wireless/mobile market and the future needs and challenges.

UNIT-I: INTRODUCTION **9**

Wireless networking trends, Key wireless physical layer concepts, Multiple access technologies, CDMA, FDMA, TDMA, Spread spectrum technologies, Frequency reuse, Radio propagation and modelling, Challenges in mobile computing: Resource poorness, Bandwidth, energy etc.

WIRELESS LOCAL AREA NETWORKS:

IEEE 802.11 Wireless LANs physical & MAC layer, 802.11 MAC Modes (DCF & PCF) IEEE 802.11 standards, Architecture & protocols, Infrastructure vs. Adhoc modes, Hidden node & Exposed terminal problem, Problems, Fading effects in indoor and outdoor WLANs, WLAN deployment issues.

UNIT-II: WIRELESS CELLULAR NETWORKS **9**

1G and 2G, 2.5G, 3G, and 4G, Mobile IPv4, Mobile IPv6, TCP over wireless networks, Cellular architecture, Frequency reuse, Channel assignment strategies, Handoff strategies, Interference and system capacity, Improving coverage and capacity in cellular systems, Spread spectrum technologies.

UNIT-III: WIRELESS SENSOR NETWORKS **9**

WiMAX (Physical layer, Media access control, Mobility and Networking), IEEE 802.22 Wireless regional area networks, IEEE 802.21 media independent Handover overview Introduction, Application, Physical, MAC layer and network layer, Power management, Tiny OS overview.

UNIT-IV: WIRELESS PANS AND SECURITY **9**

Bluetooth AND Zigbee, Introduction to wireless sensors, Security in wireless networks vulnerabilities, Security techniques, Wi– Fi Security, DoS in wireless communication.

UNIT-V: ADHOC NETWORKS **9**

IEEE 802.11x and IEEE 802.11i standards, Introduction to vehicular Adhoc networks.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Schiller J., “Mobile Communications”, Addison Wesley 2000.
2. Stallings W., “Wireless Communications and Networks”, Pearson Education 2005.

3. Stojmenic Ivan, "Handbook of Wireless Networks and Mobile Computing", John Wiley and Sons Inc 2002.
4. Yi Bing Lin and Imrich Chlamtac, "Wireless and Mobile Network Architectures", John Wiley and Sons Inc 2000.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Demonstrate advanced knowledge of networking and wireless networking and understand various types of wireless networks, standards, operations and use cases.

CO2: Design WLAN, WPAN, WWAN, Cellular based upon underlying propagation and performance analysis.

CO3: Illustrate knowledge of protocols used in wireless networks and learn simulating wireless networks.

CO4: Extend wireless networks exploring tradeoffs between wire line and wireless links.

CO5: Develop mobile applications to solve some of the real-world problems.

19CSPE209 SOFTWARE ARCHITECTURES AND DESIGN SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand the need, design approaches for software architecture to bridge the dynamic requirements and implementation.
- To learn the design principles and to apply for large scale systems.
- To design architectures for distributed heterogeneous systems, environment through brokerage interaction.
- To build design knowledge on service oriented and model driven architectures and the aspect oriented architecture.
- To develop appropriate architectures for various Case studies like semantic web services, supply chain cloud services.

UNIT-I: INTRODUCTION 9

Introduction to software architecture – Bridging requirements and implementation, Design guidelines, Software quality attributes – Software architecture design space – Agile approach to Software Architecture design – Models for software Architecture Description Languages (ADL).

UNIT-II: DESIGN PRINCIPLES FOR LARGE SCALE SYSTEMS 9

Object – Oriented paradigm – Design principles – Data – Centered software architecture: Repository architecture – Blackboard architecture – Hierarchical architecture – Main– subroutine – Master – Slave – Layered – Virtual machine – Interaction– Oriented software architectures: Model– View – Controller (MVC) – Presentation – Abstraction – Control (PAC).

UNIT-III: OVERVIEW OF DESIGN ARCHITECTURE FOR SYSTEMS HETEROGENEOUS 9

Distributed architecture: Client – Server, Middleware, Multi – Tiers, Broker Architecture – MOM, CORBA message broker architecture – Service – Oriented Architecture (SOA), SOAP, UDDI, SOA implementation in web services, Grid/cloud service computing. Heterogeneous architecture – Methodology of architecture decision, Quality attributes.

UNIT-IV: WEB SERVICE ORIENTED ARCHITECTURE 9

Architecture of user interfaces containers, Case study – Web service. Product line architectures – Methodologies, Processes and tools. Software reuse and product lines – Product line analysis, Design and implementation, Configuration models. Model driven Architectures (MDA) – Why MDA model transformation and software architecture, SOA and MDA. Eclipse modeling framework.

UNIT-V: CASE STUDIES 9

Aspect oriented architectures – AOP in UML, AOP tools, Architectural aspects and middleware selection of architectures, Evaluation of architecture designs, Case study: Online computer vendor, Order processing, Manufacture & Shipping – Inventory, Supply chain cloud service management, Semantic web services.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Ion Gorton, "Essentials of Software Architecture", 2nd Edition, Springer – Verlag, 2011.
2. Kai Qian Jones and Bartlett Publishers, "Software Architecture Design Illuminated", Canada, 2010.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Summarize the need of software architecture for sustainable dynamic systems.

CO2: Design architectures for distributed heterogeneous systems.

CO3: Have a sound knowledge on design principles and to apply for large scale systems.

CO4: Relate on service oriented and model driven architectures and the aspect-oriented architecture.

CO5: Have a working knowledge to develop appropriate architectures through various case studies.

PROFESSIONAL ELECTIVES – III

19CSPE210 **QUANTUM COMPUTING** **SEMESTER II**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To learn foundation of quantum computing.
- To understand qubits and quantum models.
- To explore quantum algorithms.
- To understand the quantum complexities.

UNIT-I: INTRODUCTION **9**

Overview of traditional computing – Church– Turing thesis – Circuit model of computation – reversible computation – Quantum physics – Quantum physics and computation – Dirac notation and Hilbert Spaces – Dual vectors – Operators – The spectral theorem – Functions of operators – Tensor products – Schmidt decomposition theorem.

UNIT-II: QUBITS AND QUANTUM MODEL OF COMPUTATION **9**

State of a quantum system – Time evolution of a closed system – Composite systems – measurement – Mixed states and general quantum operations – Quantum circuit model – quantum gates – Universal sets of quantum gates – Unitary transformations – Quantum circuits.

UNIT-III: QUANTUM ALGORITHMS-I **9**

Super dense coding – Quantum teleportation – Applications of teleportation – Probabilistic versus quantum algorithms – Phase kick– back – The Deutsch algorithm – The Deutsch– Jozsa algorithm – Simon's algorithm – Quantum phase estimation and quantum Fourier Transform – Eigen value estimation.

UNIT-IV: QUANTUM ALGORITHMS-II **9**

Order – Finding problem – Eigenvalue estimation approach to order finding – Shor's algorithm for order finding – Finding discrete logarithms – Hidden subgroups – Grover's quantum search algorithm – Amplitude amplification – Quantum amplitude estimation – Quantum counting – Searching without knowing the success probability.

UNIT-V: QUANTUM COMPUTATIONAL COMPLEXITY AND ERROR CORRECTION **9**

Computational complexity – Black-box model – Lower bounds for searching – General black – Box lower bounds – Polynomial method – Block sensitivity – Adversary methods – Classical error correction – Classical three - bit code – Fault tolerance – Quantum error correction – Three - and nine-qubit quantum codes – Fault-tolerant quantum computation.

Contact Periods:

Lecture: 45 Periods **Tutorial: 0 Periods** **Practical: 0 Periods** **Total: 45 Periods**

REFERENCES

1. Kaye P, Laflamme R, and Mosca M., "An introduction to Quantum Computing", Oxford University Press, 1999.
2. Sahni V., "Quantum Computing", Tata McGraw– Hill Publishing Company, 2007.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Apply Quantum computing Techniques.

CO2: Design Quantum models.

CO3: Implement Quantum Algorithms.

CO4: Compute the Quantum complexities.

19CSPE211 REAL TIME OPERATING SYSTEMS SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To learn real time operating system concepts, the associated issues and techniques.
- To understand design and synchronization problems in real time system.
- To explore the concepts of real time databases.
- To understand the evaluation techniques, present in real time system.

UNIT-I: REAL TIME SYSTEM AND SCHEDULING 9

Introduction– Structure of a real time system –Task classes – Performance measures for real Time systems – Estimating program run times – Issues in real time computing – Task assignment and scheduling – Classical uniprocessor scheduling algorithms – Fault tolerant scheduling.

UNIT-II: SOFTWARE REQUIREMENTS ENGINEERING 9

Requirements engineering process – Types of requirements – Requirements specification for real time systems – Formal methods in software specification – Structured analysis and design – Object oriented analysis and design and unified modelling language – Organizing the requirements document – Organizing and writing documents – Requirements validation and revision.

UNIT-III: INTERTASK COMMUNICATION AND MEMORY MANAGEMENT 9

Buffering data – Time relative Buffering– Ring Buffers – Mailboxes – Queues – Critical regions – Semaphores – Other Synchronization mechanisms – Deadlock – Priority inversion – Process stack management – Run time ring buffer – Maximum stack size multiple stack arrangement – Memory management in task control block – Swapping – Overlays – Block page management – Replacement algorithms – Memory locking – Working sets – Real time garbage collection – Contiguous file systems.

UNIT-IV: REAL TIME DATABASES 9

Real time databases – Basic definition, Real time Vs General purpose databases, Main memory databases, Transaction priorities, Transaction aborts, Concurrency control issues, Disk scheduling algorithms, Two-phase approach to improve predictability – Maintaining serialization consistency – Databases for hard real time systems.

UNIT-V: EVALUATION TECHNIQUES AND CLOCK SYNCHRONIZATION 9

Reliability evaluation techniques – Obtaining parameter values, Reliability models for hardware redundancy – Software error models. Clock synchronization – Clock, A non fault– tolerant synchronization algorithm – Impact of faults – Fault tolerant synchronization in hardware – Fault tolerant synchronization in software.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Krishna C M, Kang G. Shin, "Real– Time Systems", McGraw– Hill International Editions, 1997.
2. Philip. A. Laplante, "Real Time System Design and Analysis", Prentice Hall of India, 3rd Edition, 2004.

3. Rajib Mall, “Real– time systems: theory and practice”, Pearson Education, 2009.
4. Buhur R J A and Bailey D.L.,, “An Introduction to Real – Time Systems”, Prentice Hall International, 1999.
5. Stuart Bennett, “Real Time Computer Control – An Introduction”, Prentice Hall of India, 1998.
6. Allen Burns and Andy Wellings, “Real Time Systems and Programming Languages”, Pearson Education, 2003.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Apply principles of real time system design techniques to develop real time applications.
CO2: Make use of database in real time applications.
CO3: Discuss architectures and behavior of real time operating systems.
CO4: Design the real time database.
CO5: Show evaluation techniques in application.

19CSPE212 **SOFTWARE QUALITY ASSURANCE AND TESTING** **SEMESTER II**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand the basics of testing, test planning &design and test team organization.
- To study the various types of test in the life cycle of the software product.
- To build design concepts for system testing and execution.
- To learn the software quality assurance, metrics, defect prevention techniques.
- To learn the techniques for quality assurance and applying for applications.

UNIT-I: SOFTWARE TESTING - CONCEPTS, ISSUES, AND TECHNIQUES 9

Quality revolution, Verification and validation, Failure, Error, Fault, and Defect, Objectives of testing, Testing activities, Test case selection white – box and black, Test planning and design, Test tools and automation, Power of test. Test team organization and management –Test groups, Software quality assurance group, System test team hierarchy, Team building.

UNIT-II: SYSTEM TESTING 9

System testing – System integration techniques – Incremental, Top down bottom up sandwich and big bang, Software and hardware integration, Hardware design verification tests, Hardware and software compatibility matrix test plan for system integration. Built – in testing. functional testing – Testing a function in context. Boundary value analysis, Decision tables. acceptance testing – Selection of acceptance criteria, Acceptance test plan, Test execution Test. Software reliability – Fault and failure, Factors influencing software, Reliability models.

UNIT-III: SYSTEM TEST CATEGORIES 9

System test categories taxonomy of system tests, Interface tests functionality tests. GUI tests, Security tests feature Tests, Robustness tests, Boundary value tests power cycling tests Interoperability tests, Scalability tests, Stress tests, Load and stability tests, Reliability tests, Regression tests, Regulatory tests. Test generation from FSM models – State– Oriented Model. Finite – state machine transition tour method, Testing with state verification. Test architectures– Local, Distributed, Coordinated, Remote. System test design – Test design Factors requirement identification, Modeling a test design process test design preparedness, Metrics, Test case design effectiveness. System test execution – Modeling defects, Metrics for monitoring test execution. Defect reports, Defect causal analysis, Beta testing, Measuring test effectiveness.

UNIT-IV: SOFTWARE QUALITY 9

Software quality - People‘s Quality Expectations, Frameworks and ISO-9126, McCall‘s Quality Factors and Criteria – Relationship. Quality Metrics. Quality Characteristics ISO 9000:2000 Software Quality Standard. Maturity models- Test Process Improvement ,Testing Maturity Model.

UNIT-V: SOFTWARE QUALITY ASSURANCE 9

Quality Assurance – Root Cause Analysis, modeling, technologies, standards and methodologies for defect prevention. Fault Tolerance and Failure Containment – Safety Assurance and Damage Control, Hazard analysis using fault– trees and event– trees. Comparing Quality Assurance Techniques and Activities. QA Monitoring and Measurement, Risk

Identification for Quantifiable Quality Improvement. Case Study: FSM– Based Testing of Web–Based Applications.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Kshirasagar Nak Priyadarshini Tripathy, "Software Testing and Quality Assurance – Theory and Practice", John Wiley & Sons Inc,2008.
2. Jeff Tian, "Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement", John Wiley & Sons, Inc., Hoboken, New Jersey. 2005.
3. Daniel Galin, "Software Quality Assurance – From Theory to Implementation", Pearson Education Ltd UK, 2004.
4. Milind Limaye, "Software Quality Assurance", TMH , New Delhi, 2011.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Perform functional and nonfunctional tests in the life cycle of the software product.

CO2: Rephrase system testing and test execution process.

CO3: Categories System testing.

CO4: Identify defect prevention techniques and software quality assurance metrics.

CO5: Apply techniques of quality assurance for typical applications.

19CSPE213 ADVANCED MICROCONTROLLERS AND APPLICATIONS IN EMBEDDED SYSTEMS

SEMESTER II

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand the architecture of embedded processor, microcontroller and peripheral devices.
- To interface memory and peripherals with embedded systems.
- To study the embedded network environment.
- To understand challenges in Real time operating systems.
- To study, analyze and design applications on embedded systems.

UNIT-I: EMBEDDED PROCESSORS 9

Embedded Computers – Characteristics of Embedded Computing Applications – Challenges in Embedded Computing System Design – Embedded System Design Process – Formalism for System Design – Structural Description – Behavioral Description – ARM Processor – Intel ATOM Processor.

UNIT-II: EMBEDDED COMPUTING PLATFORM 9

CPU Bus Configuration – Memory Devices and Interfacing – Input/Output Devices and Interfacing – System Design – Development and Debugging – Emulator – Simulator – JTAG Design Example – Alarm Clock – Analysis and Optimization of Performance – Power and Program Size.

UNIT-III: EMBEDDED NETWORK ENIVIRONMENT 9

Distributed embedded architecture – Hardware and software architectures – Networks for embedded systems – I2C – CAN Bus – SHARC Link Supports – Ethernet – Myrinet – Internet – Network – based design – Communication analysis – System performance analysis – Hardware platform design – Allocation and Scheduling – Design example – Elevator controller.

UNIT-IV: REAL– TIME CHARACTERISTICS 9

Clock driven approach – Weighted round robin approach – Priority driven approach – Dynamic versus static systems – Effective release times and deadlines – Optimality of the Earliest Deadline First (EDF) Algorithm – Challenges in validating timing constraints in Priority driven systems – Off– Line versus On – line scheduling.

UNIT-V: SYSTEM DESIGN TECHNIQUES 9

Design methodologies – Requirement analysis – Specification – System analysis and architecture design – Quality assurance – Design examples – Telephone PBX – Ink jet printer – Personal digital assistants – Set– Top boxes.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Adrian McEwen, "Designing the Internet of Things" Wiley Publication, Ist Edition, 2013.
2. Andrew N. Sloss, D. Symes, C. Wright, "Arm System Developers Guide", Morgan Kauffman/Elsevier, 2006.
3. Arshdeep Bahga, Vijay Madisetti, " Internet of Things: A Hands-on-Approach", VPT Ist Edition, 2014.

4. Krishna C M and Shin K G., “Real– Time Systems” , McGraw– Hill, 1997.
5. Frank Vahid and Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, John Wiley & Sons.
6. Jane W.S Liu, “Real-Time Systems”, Pearson Education Asia.
7. Steve Heath, “Embedded System Design”, Elsevier, 2005.
8. Wayne Wolf, “Computers as Components: Principles of Embedded Computer System Design”, Elsevier, 2006.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Outline the architecture of embedded processor, microcontroller and peripheral devices.
CO2: Interface memory and peripherals with embedded systems.
CO3: Implement embedded network environment.
CO4: Know the real time characters of Real time operating systems.
CO5: Know the design techniques for Real time operating systems.

19CSPE214 SOCIAL NETWORKS AND ANALYSIS SEMESTER II

L	T	P	C
3	**0**	**0**	**3**

COURSE OBJECTIVES:

- To understand the components of the social network.
- To model and visualize the social network.
- To mine the users in the social network.
- To understand the evolution of the social network.
- To know the applications in real time systems.

UNIT-I: INTRODUCTION 9

Introduction to Web –` Limitations of current web – Development of semantic web – Emergence of the social web – Statistical properties of social networks – Network analysis – Development of social network analysis – Key concepts and measures in network analysis – Discussion networks – Blogs and online communities – Web – Based networks.

UNIT-II: MODELING AND VISUALIZATION 9

Visualizing online social networks – A Taxonomy of visualizations – Graph representation – Centrality – Clustering – Node-Edge diagrams – Visualizing social networks with matrix – Based representations – Node-Link diagrams – Hybrid representations – Modelling and aggregating social network data – Random Walks and their Applications – Use of Hadoop and Map Reduce – Ontological representation of social individuals and relationships.

UNIT-III: MINING COMMUNITIES 9

Aggregating and reasoning with social network data, Advanced representations – Extracting evolution of web community from a series of web archive – Detecting communities in social networks – Evaluating communities – Core methods for community detection & mining – Applications of community mining algorithms – Node classification in social networks.

UNIT-IV: EVOLUTION 9

Evolution in social networks – Framework – Tracing smoothly evolving communities – Models and algorithms for social influence analysis – Influence related statistics – Social similarity and influence – Influence maximization in viral marketing – Algorithms and systems for expert location in social networks – Expert location without graph constraints – with score propagation – Expert team formation – Link prediction in social networks – Feature based link prediction – Bayesian probabilistic models – Probabilistic relational models.

UNIT-V: APPLICATIONS 9

A Learning based approach for real time emotion classification of tweets – A new linguistic approach to assess the opinion of users in social network environments – Explaining scientific and technical emergence forecasting – Social network analysis for biometric template protection.

Contact Periods:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Ajith Abraham, Aboul Ella Hassanien, Vaclav Sna.el, “Computational Social Network Analysis: Trends, Tools and Research Advances”, Springer, 2012.

2. Borko Furht, “Handbook of Social Network Technologies and Applications”, Springer, 1st Edition, 2011.
3. Charu C. Aggarwal, “Social Network Data Analytics”, Springer; 2014.
4. Giles, Mark Smith, John Yen, “Advances in Social Network Mining and Analysis”, Springer, 2010.
5. Guandong Xu, Yanchun Zhang and Lin Li, “Web Mining and Social Networking Techniques and applications”, Springer, 1st Edition, 2012.
6. Peter Mika, Social Networks and the Semantic Web., Springer, 1st Edition, 2007.
7. Przemyslaw Kazienko, Nitesh Chawla, “Applications of Social Media and Social Network Analysis”, Springer, 2015.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Work on the internals components of the social network.

CO2: Model and visualize the social network.

CO3: Mine the behavior of the users in the social network.

CO4: Predict the possible next outcome of the social network.

CO5: Design the application of social network

19CSPE215 **DATA PREPARATION AND ANALYSIS** **SEMESTER II**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To prepare the data for analysis and develop meaningful data visualizations.
- To know about the Data Cleaning and Exploratory Analysis.
- To learn about different data visualization techniques

UNIT-I: DATA GATHERING AND PREPARATION **9**

Consistency checking, Heterogeneous and missing data, Data transformation and segmentation.

UNIT-II: DATA CLEANING **9**

Consistency checking, Heterogeneous and missing data, Data transformation and segmentation.

UNIT-III: EXPLORATORY ANALYSIS **9**

Descriptive and comparative statistics, Clustering and association, Hypothesis generation.

UNIT-IV: VISUALIZATION **9**

Designing visualizations, Time series, Geo located data, Correlations and connections, Hierarchies and networks, Interactivity.

UNIT-V: DATA ANALYSIS **9**

Structure of information systems, Basics of IS planning, Statistical analysis of data, Numeric and linguistic data, Data clustering, Statistical models for data processing, Data warehouse, OLAP, Data mining.

Contact Periods:

Lecture: 45 Periods **Tutorial: 0 Periods** **Practical: 0 Periods** **Total: 45 Periods**

REFERENCES

1. Making sense of Data : A practical Guide to Exploratory Data Analysis and Data Mining, by Glenn.
2. Exploratory Data Mining and Data Cleaning, by Tamraparni Dasu.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Gather the data for performing the analysis.
CO2: Clean the data.
CO3: Generate statistics.
CO4: Extract the data for performing the analysis.

19CSPE216	SECURE SOFTWARE DESIGN & ENTERPRISE COMPUTING	SEMESTER II			
		L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- To provide software development perspective to the challenges of engineering software systems that are secure.
- To design and implementation issues critical to producing secure software systems.
- To make the requirements for confidentiality, integrity, and availability integral to the software development process from requirements gathering to design, development, configuration, deployment, and ongoing maintenance.

UNIT-I: INTRODUCTION 9

Defining computer security, The principles of secure software, Trusted computing base, etc, threat modeling, Advanced techniques for mapping security requirements into design specifications. Secure software implementation, Deployment and ongoing management.

UNIT-II: SOFTWARE DESIGN AND SECURITY 9

Software design and an introduction to hierarchical design representations. Difference between high level and detailed design. Handling security with high level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, Quality assurance activities and strategies that support early vulnerability detection, Trust models, security Architecture & design reviews.

UNIT-III: SOFTWARE ASSURANCE AND TESTING 9

Software assurance model: Identify project security risks & selecting risk management strategies, Risk management framework, Security best practices/ Known security flaws, Architectural risk analysis, Security testing & reliability (Penn testing, Risk – based security Testing, Abuse cases, Operational testing, Introduction to reliability engineering, Software reliability, Software reliability approaches, Software reliability modeling.

UNIT-IV: SOFTWARE IN ENTERPRISE 9

Software Security in Enterprise Business: Identification and authentication, Enterprise Information Security, Symmetric and asymmetric cryptography, Including public key cryptography, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Algorithms for hashes and message digests. Authentication, Authentication schemes, Access control models, Kerberos protocol, Public Key Infrastructure (PKI), Protocols specially designed for e – commerce and web applications, Firewalls and VPNs. Management issues, technologies and systems related to information security management at enterprises.

UNIT-V: WORKING WITH FRAMEWORKS 9

Security development frameworks. Security issues associated with the development and deployment of information systems, including Internet– based e-commerce, e-business, and e-service systems, as well as the technologies required to develop secure information systems for enterprises, Policies and regulations essential to the security of enterprise information systems.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Stallings W., “Cryptography and network security: Principles and practice”, 5th Edition, Upper Saddle River, NJ: Prentice Hall., 2011.
2. Kaufman C, Perlman R and Speciner M., “Network security: Private communication in a public world”, 2nd Edition, Upper Saddle River, NJ:Prentice HalL, 2002.
3. Pfleeger C P and Pfleeger S L., “Security in Computing”, 4th Edition, Upper Saddle River, NJ:Prentice Hall, 2007.
4. Merkow M, and Breithaupt J., “Information security: Principles and Practices. Upper Saddle River”, NJ:Prentice Hall, 2005.
5. Gary McGraw, “Software Security: Building Security In”, Addison– Wesley, 2006.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Recognize various aspects and principles of software security.

CO2: Devise security models for implementing at the design level.

CO3: Identify and analyze the risks associated with Software engineering and use relevant models to mitigate the risks.

CO4: Relate the various security algorithms to implement secured computing and computer networks.

CO5: Explain different security frameworks for different types of systems including electronic systems.

19CSPE217 PERFORMANCE ANALYSIS OF COMPUTER SYSTEMS — **SEMESTER I**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To understand the mathematical foundations needed for performance evaluation of computer systems.
- To understand the metrics used for performance evaluation.
- To understand the analytical modeling of computer systems.
- To enable the students to develop new queuing analysis for both simple and complex systems.
- To appreciate the use of smart scheduling and introduce the students to analytical techniques for evaluating scheduling policies.

UNIT-I: OVERVIEW OF PERFORMANCE EVALUATION **9**

Need for performance evaluation in computer systems – Overview of performance evaluation methods – Introduction to queuing – probability review – Generating random variables for simulation – Sample paths, Convergence and averages – Little 's law and other operational Laws – Modification for closed Systems.

UNIT-II: MARKOV CHAINS AND SIMPLE QUEUES **9**

Discrete– Time markov chains – Ergodicity theory – Real world examples – Google, Aloha – Transition to continuous – Time markov chain – M/M/1.

UNIT-III: MULTI– SERVER AND MULTI– QUEUE SYSTEMS **9**

Server farms: M/M/k and M/M/k/k – Capacity provisioning for server farms – Time reversibility and Burke 's theorem – Networks of queues and Jackson product form – Classed and closed networks of queues.

UNIT-IV: REAL– WORLD WORKLOADS **9**

Case study of real– world workloads – Phase– Type distributions and matrix – Analytic methods – Networks with time– Sharing servers – M/G/1 queue and the inspection paradox – Task assignment policies for server farms.

UNIT-V: SMART SCHEDULING IN THE M/G/1 **9**

Performance metrics – Scheduling Non – preemptive and Preemptive non – Size – Based policies – Scheduling Non – preemptive and preemptive size – Based policies – Scheduling – SRPT and Fairness.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Trivedi K S., "Probability and Statistics with Reliability, Queueing and Computer Science Applications", John Wiley and Sons, 2001.
2. Krishna Kant, "Introduction to Computer System Performance Evaluation", McGraw– Hill, 1992.
3. Lieven Eeckhout, "Computer Architecture Performance Evaluation Methods", Morgan and Claypool Publishers, 2010.

4. Mor Harchol – Balter, “Performance Modeling and Design of Computer Systems – Queueing Theory in Action, Cambridge University Press, 2013.
5. Paul J. Fortier and Howard E. Michel, “Computer Systems Performance Evaluation and Prediction”, Elsevier, 2003.
6. Raj Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling”, Wiley– Interscience, 1991.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Identify the need for performance evaluation and the metrics used for it.

CO2: Distinguish between open and closed queuing networks.

CO3: Apply the operational laws to open and closed systems.

CO4: Use discrete time and continuous time Markov chains to model real world systems.

CO5: Develop analytical techniques for evaluating scheduling policies.

19CSPE218	MULTI-CORE ARCHITECTURES AND PROGRAMMING	SEMESTER I
		L T P C
		3 0 0 3

COURSE OBJECTIVES:

- To understand the need for multi-core processors, and their architecture.
- To understand the challenges in parallel and multi-threaded programming.
- To learn about the various parallel programming paradigms.
- To develop multicore programs and design parallel solutions.

UNIT-I: MULTI– CORE PROCESSORS **9**

Single core to Multi – Core architectures – SIMD and MIMD systems – Interconnection networks – Symmetric and distributed shared memory architectures – Cache coherence – Performance Issues – Parallel program design.

UNIT-II: PARALLEL PROGRAM CHALLENGES **9**

Performance – Scalability – Synchronization and data sharing – Data races – Synchronization primitives (mutexes, locks, semaphores, barriers) – Deadlocks and livelocks – Communication between threads (condition variables, signals, message queues and pipes).

UNIT-III: SHARED MEMORY PROGRAMMING WITH OPENMP **9**

Open MP execution model – Memory model – Open MP directives – Work – sharing Constructs – Library functions – Handling data and functional parallelism – Handling loops – Performance considerations.

UNIT-IV: DISTRIBUTED MEMORY PROGRAMMING WITH MPI **9**

MPI program execution – MPI constructs – libraries – MPI send and receive – Point– to– point and Collective communication – MPI derived datatypes – Performance evaluation.

UNIT-V: PARALLEL PROGRAM DEVELOPMENT **9**

Case studies – n– Body solvers – Tree search – Open MP and MPI implementations and comparison.

Contact Periods:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES

1. Peter S. Pacheco, “An Introduction to Parallel Programming”, Morgan–Kauffman/Elsevier, 2011.
2. Darryl Gove, “Multicore Application Programming for Windows, Linux, and Oracle Solaris”, Pearson, 2011 (unit 2).
3. Michael J. Quinn, “Parallel programming in C with MPI and OpenMP”, Tata McGraw Hill, 2003.
4. Victor Alessandrini, “Shared Memory Application Programming”, 1st Edition, Concepts and Strategies in Multicore Application Programming, Morgan Kaufmann, 2015.
5. Yan Solihin, “Fundamentals of Parallel Multicore Architecture”, CRC Press, 2015.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Describe multicore architectures and identify their characteristics and challenges.

CO2: Identify the issues in programming parallel processors.

CO3: Write programs using OpenMP and MPI.

CO4: Design parallel programming solutions to common problems.

CO5: Compare programming for serial processors and programming for parallel.

PROFESSIONAL ELECTIVES – IV

19CSPE301 CYBER SECURITY AND DIGITAL FORENSICS SEMESTER III

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To identify the difference between threat, risk, attack and vulnerability.
- To build protection mechanisms in order to secure operating systems and computer networks.
- To apply methods for authentication, access control, intrusion detection and prevention.
- To develop knowledge in forensics and privacy issues.
- To identify and mitigate software security vulnerabilities in existing systems.

UNIT-I: INTRODUCTION TO CYBER SECURITY 9

Introduction, Security Goals, Attacks, Services and Mechanisms, Understanding Threats and Vulnerabilities, Authentication and Access Control.

UNIT-II: OPERATING SYSTEM & NETWORKS SECURITY 9

Security in Operating Systems, Memory and Address protection, Rootkit, File protection mechanisms, Trusted Operating Systems design, Network security attack, Threats to Network Communications, Wireless Network Security, Denial of Service, Distributed Denial-of-Service.

UNIT-III: SECURITY COUNTERMEASURES 9

Cryptography in Network Security, Firewalls, Intrusion Detection and Prevention Systems, Network Management, Databases, Security Requirements of Databases, Reliability and Integrity, Database Disclosure, Data Mining and Big Data.

UNIT-IV: COMPUTER FORENSICS 9

Forensics Overview, Benefits of Computer Forensics, Computer Crimes, Computer Forensics Evidence and the Courts, Legal Concerns and Privacy Issues.

UNIT-V: PLANNING AND INCIDENTS 9

Security Planning, Business Continuity Planning, Handling Incidents, Risk Analysis, Dealing with Disaster, Emerging Technologies, The Internet of Things, Economics, Electronic Voting, Cyber Warfare, Cyberspace and the Law, International Laws, Cybercrime, Cyber Warfare and Home Land Security.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

3. Charles P. Pfleeger Shari Lawrence Pfleeger Jonathan Margulies, “Security in Computing”, 5th Edition, Pearson Education, 2015.

4. Behrouz A Forouzan, Debdeep Mukhopadhyay, “Cryptography and Network Security”, Tata Mc-Graw Hill, 2010.
5. George K. Kostopoulous, “Cyber Space and Cyber Security”, CRC Press, 2013.
6. Martti Lehto, Pekka Neittaanmaki, “Cyber Security: Analytics, Technology and Automation”, Springer International Publishing, Switzerland, 2015.

7. Nelson Phillips and Enfinger Steuart, “Computer Forensics and Investigations”, Cengage Learning, New Delhi, 2009.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the cryptography concepts and applications.
CO2: Design a secure operating system.
CO3: Compare the various security measures.
CO4: Define the laws relevant to computer forensics.
CO5: Incorporate the approaches for incident analysis and response.

19CSPE302 **OPEN SOURCE SOFTWARE** **SEMESTER III**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- Be exposed to the context and operation of free and open source software communities and associated software projects.
- To be familiar with participating in a FOSS project
- To learn scripting and programming languages.
- Be familiar with participating in a FOSS project.

UNIT-I: PHILOSOPHY **9**

Linux, GNU and Freedom, Brief history of GNU, licensing free software – GPL and copy Left, trends and potential – global and Indian, overview and usage of various Linux Distributions – user friendliness perspective – scientific perspective.

UNIT-II: LINUX **9**

Linux Installation and Hardware Configuration – Boot Process-The Linux Loader (LILO) – The Grand Unified Bootloader (GRUB) – Dual-Booting Linux and other Operating System – Boot-Time Kernel Options – X Windows System Configuration –System Administration – Backup and Restore Procedures – Strategies for keeping a Secure Server.

UNIT-III: FOSS PROGRAMMING PRACTICES **9**

GNU debugging tools, using source code versioning and managing tools, Review of common programming practices and guidelines for GNU/Linux and FOSS, Documentation.

UNIT-IV: PROGRAMMING TECHNIQUES **9**

Application programming – Basics of X Windows server architecture – QT programming – GTK + Programming – Python programming – Open source equivalent of existing Commercial Software.

UNIT-V: PROJECTS AND CASE STUDIES **9**

Linux for portable Devices, Creation of Bootable CD and USB from command line, Case Studies – Samba, Libre office, Assistive technology.

CONTACT PERIODS:

Lecture: 45 Periods **Tutorial: 0 Periods** **Practical: 0 Periods** **Total: 45 Periods**

REFERENCES:

1. Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins, “Linux in a Nutshell”, Sixth Edition, OReilly Media, 2009.
2. Philosophy of GNU URL: http://www.gnu.org/philosophy/
3. Overview of Linux Distributions URL: http://distrowatch.com/dwres.php?resource=major
4. Introduction to Linux – A Hands on Guide, URL: http://tldp.org/guides.html
5. Case study., Libre office: http://www.libreoffice.org/

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Install and run open-source operating systems.

CO2: Gather information about Free and Open Source Software projects from software releases and from sites on the internet.
CO3: Build and modify one or more Free and Open Source Software packages.
CO4: Use a version control system.
CO5: Contribute software to and interact with Free and Open Source Software development projects.

19CSPE303 BLOCK CHAIN TECHNOLOGY SEMESTER III

L	T	P	C
3	**0**	**0**	**3**

COURSE OBJECTIVES:

- Learn block chain systems mainly Bitcoin and Ethereum work.
- To interact securely.
- Design, build and deploy smart contracts and distributed applications.
- Integrate ideas from block chain technology into their own projects.

UNIT-I: BASICS 9

Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete. Cryptography: Hash function, Digital Signature – ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.

UNIT-II: BLOCKCHAIN 9

Introduction, Advantage over conventional distributed database, Block chain Network, Mining Mechanism, Distributed Consensus, Merkle Patricia Tree, Gas Limit, Transactions and Fee, Anonymity, Reward, Chain Policy, Life of Block chain application, Soft & Hard Fork, Private and Public block chain.

UNIT-III: DISTRIBUTED CONSENSUS 9

Nakamoto consensus, Proof of Work, Proof of Stake, Proof of Burn, Difficulty Level, Sybil Attack, Energy utilization and alternate.

UNIT-IV: CRYPTOCURRENCY 9

History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum - Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin

UNIT-V: CRYPTOCURRENCY REGULATION 9

Stakeholders, Roots of Bit coin, Legal Aspects-Crypto currency Exchange, Black Market and Global Economy. Applications: Internet of Things, Medical Record Management System, Domain Name Service and future of Blockchain.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, July 19, 2016.
2. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'reilly,2010.
3. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System.
4. Gavin Wood, "ETHEREUM: A Secure Decentralized Transaction Ledger", Yellow paper.2014.
5. Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli, A survey of attacks on Ethereum smart contracts.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain design principles of Bitcoin.

CO2: Explain the Simplified Payment Verification protocol.

CO3: List and describe differences between proof-of-work and proof-of-stake consensus.

CO4: Interact with a block chain system by sending and reading transactions.

CO5: Design, build, and deploy a distributed application and Evaluate security, privacy and efficiency of a given block chain system.

19CSPE304 ROBOTICS PROCESS AUTOMATION SEMESTER III

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To develop the student's knowledge in various robot structures and their workspace.
- To develop student's skills in performing spatial transformations associated with rigid body motions and robot systems.
- To provide the student with knowledge of the singularity issues associated with the operation of robotic systems.

UNIT-I: INTRODUCTION 9

History of robots, Classification of robots, Present status and future trends. Basic components of robotic system. Basic terminology– Accuracy, Repeatability, Resolution, Degree of freedom. Mechanisms and transmission, End effectors, Grippers-different methods of gripping, Mechanical Grippers-Slider crank mechanism, Screw type, Rotary actuators, Cam type gripper, Magnetic grippers, Vacuum grippers, Air operated grippers; Specifications of robot.

UNIT-II: DRIVE SYSTEMS AND SENSORS 9

Drive system- hydraulic, pneumatic and electric systems Sensors in robot – Touch sensors, Tactile sensor, Proximity and range sensors, Robotic vision sensor, Force sensor, Light sensors, Pressure sensors.

UNIT-III: KINEMATICS AND DYNAMICS OF ROBOTS 9

2D, 3D Transformation, Scaling, Rotation, Translation, Homogeneous coordinates, multiple transformation, Simple problems. Matrix representation, Forward and Reverse Kinematics of Three Degree of Freedom, Homogeneous Transformations, Inverse kinematics of Robot, Robot Arm dynamics, D-H representation of robots, Basics of Trajectory Planning.

UNIT-IV: ROBOT CONTROL 9

Robot Controls-Point to point control, Continuous path control, Intelligent robot, Control system for robot joint, Control actions, Feedback devices, Encoder, Resolver, LVDT, Motion Interpolations, Adaptive control.

UNIT-V: PROGRAMMING AND APPLICATIONS 9

Introduction to Robotic Programming, On-line and off-line programming, programming examples. Robot Applications-Material handling, Machine loading and unloading, assembly, Inspection, Welding, Spray painting.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Mikell P Groover, Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, "Industrial Robotics, Technology programming and Applications", McGraw Hill, 2012.
2. Craig. J. J. "Introduction to Robotics- mechanics and control", Addison- Wesley, 1999.
3. S.R. Deb, "Robotics Technology and flexible automation", Tata McGraw-Hill Education., 2009.
4. Richard D. Klafter, Thomas .A, ChriElewski, Michael Negin, "Robotics Engineering an Integrated Approach", PHI Learning., 2009.

5. Francis N. Nagy, Andras Siegler, "Engineering foundation of Robotics", Prentice Hall Inc., 1987.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Ability to knowledge in various robot structures and their workspace.

CO2: Deep knowledge in Drive system and Sensor applications.

CO3: Ability to solve inverse kinematics of simple robot manipulators.

CO4: Explain about the control system concept in robotics.

CO5: Reason from definitions to construct mathematical proofs.

19CSPE305 ADVANCED SOFTWARE ENGINEERING SEMESTER III

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To Learn Software Engineering Lifecycle Models.
- To do project management and cost estimation.
- To gain knowledge of the System Analysis and Design concepts.
- To gain knowledge of software testing approaches.
- To be familiar with DevOps practices.

UNIT-I: INTRODUCTION 9

Software engineering concepts – Development activities – Software lifecycle models – Classical waterfall – Iterative waterfall – Prototyping – Evolutionary – Spiral – Software project management – Project planning – Estimation – Scheduling – Risk management – Software configuration management.

UNIT-II: SOFTWARE REQUIREMENT SPECIFICATION 9

Requirement analysis and specification – Requirements gathering and analysis – Software Requirement Specification – Formal system specification – Finite State Machines – Petrinets – Object modelling using UML – Use case Model – Class diagrams – Interaction diagrams – Activity diagrams – State chart diagrams – Functional modelling – Data Flow Diagram.

UNIT-III: ARCHITECTURE AND DESIGN 9

Software design – Design process – Design concepts –Coupling – Cohesion – Functional independence – Design patterns – Model-view-controller – Publish-subscribe – Adapter – Command – Strategy – Observer – Proxy – Facade – Architectural styles – Layered – Client-server -Tiered – Pipe and filter – User interface design.

UNIT-IV: TESTING 9

Testing – Unit testing – Black box testing – White box testing – Integration and System testing – Regression testing – Debugging – Program analysis – Symbolic execution – Model Checking.

UNIT-V: DevOps 9

DevOps: Motivation – Cloud as a Platform-Operations – Deployment Pipeline: Overall Architecture – Building and Testing – Deployment – Case study: Migrating to Micro services.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Bernd Bruegge, Alan H Dutoit, Object-Oriented Software Engineering, 2nd edition, Pearson Education, 2004.
2. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, 2nd edition, PHI Learning Pvt. Ltd., 2010.
3. Craig Larman, Applying UML and Patterns, 3rd ed, Pearson Education, 2005.
4. Len Bass, Ingo Weber and Liming Zhu, DevOps: A Software Architect‘s Perspective, Pearson Education, 2016
5. Rajib Mall, Fundamentals of Software Engineering, 3rd edition, PHI Learning Pvt. Ltd., 2009.

6. Stephen Schach, Software Engineering 7th edition, McGraw-Hill, 2007.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the advantages of various Software Development Lifecycle Mode.

CO2: Gain knowledge on project management approaches as well as cost and schedule estimation strategies.

CO3: Perform formal analysis on specifications.

CO4: Use UML diagrams for analysis and design.

CO5: Describe the advantages of DevOps practices.

19CSPE306 **GPU COMPUTING** **SEMESTER III**

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To learn about the basics of programming for heterogeneous architectures.
- To know programming for massively parallel processors.
- To identify the issues in mapping algorithms for GPUs.
- To introduce different GPU programming models.

UNIT-I: GPU ARCHITECTURE **9**

Understanding Parallelism with GPU – Typical GPU Architecture – CUDA Hardware Overview – Threads, Blocks, Grids, Warps, Scheduling – Memory Handling with CUDA: Shared Memory, Global Memory, Constant Memory and Texture Memory.

UNIT-II: GPU PROGRAMMING **9**

Using CUDA – Multi GPU – Multi GPU Solutions – Optimizing CUDA Applications: Problem Decomposition, Memory Considerations, Transfers, Thread Usage, Resource Contentions, Self-tuning Applications.

UNIT-III: PROGRAMMING ISSUES **9**

Common Problems: CUDA Error Handling, Parallel Programming Issues, Synchronization, Algorithmic Issues, Finding and Avoiding Errors.

UNIT-IV: ALGORITHMS ON GPU **9**

Parallel Patterns: Convolution, Prefix Sum, Sparse Matrix – Matrix Multiplication – Programming Heterogeneous Cluster – CUDA Dynamic Parallelism.

UNIT-V: OTHER GPU PROGRAMMING MODELS **9**

Introducing OpenCL, OpenACC, Thrust.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Shane Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing), First Edition, Morgan Kaufmann, 2012.
2. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors - A Hands-on Approach, Second Edition, Morgan Kaufmann, 2012.
3. Nicholas Wilt, CUDA Handbook: A Comprehensive Guide to GPU Programming, Addison - Wesley, 2013.
4. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General Purpose GPU Programming, Addison - Wesley, 2010.
5. http://www.nvidia.com/object/cuda_home_new.html

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Describe GPU Architecture.

CO2: Write programs using CUDA.

CO3: Implement algorithms in GPUs to get maximum occupancy and throughput.

CO4: Program in any heterogeneous programming model.

19CSPE307	ADVANCED OPTIMIZATION TECHNIQUES	SEMESTER III			
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

COURSE OBJECTIVES:

- To acquire knowledge to adopt linear programing.

- Enrich the idea of numerical optimization methods.
- Explore working principle of genetic algorithms.
- Develop genetic programming for solving equations.

UNIT-I: LINEAR PROGRAMMING AND ASSIGNMENT PROBLEM 9

Linear Programming –Two-phase simplex method, Big-M method, duality, interpretation, applications, Assignment problem- Hungarian's algorithm, Degeneracy, applications, unbalanced problems, traveling salesman problem.

UNIT-II: CLASSICAL AND NUMERICAL OPTIMIZATION TECHNIQUES 9

Classical optimization Techniques –Single variable optimization with and without constraints, multi–variable, optimization without constraints, multi–variable optimization with constraints– method of Lagrange multipliers, Kuhn-Tucker conditions. Numerical methods for optimization- Nelder Mead's Simplex search method, Gradient of a function, Steepest descent method, Newton's method, types of penalty methods for handling constraints.

UNIT-III: GENETIC ALGORITHM 9

Genetic algorithm (GA) –Differences and similarities between conventional and evolutionary algorithms, working principle, reproduction, crossover, mutation, termination criteria, different reproduction and crossover operators, GA for constrained optimization, draw backs of GA.

UNIT-IV: GENETIC PROGRAMMING 9

Genetic Programming (GP) –Principles of genetic programming, terminal sets, functional sets, differences between GA & GP, random population generation, solving differential equations using GP.

UNIT-V: MULTI-OBJECTIVE GA 9

Multi-ObjPareto's analysis, Non-dominated front, multi–objective GA, Non-dominated sorted GA, convergence criterion, applications of multi-objective problems Basic Problem solving using Genetic algorithm, Genetic Programming & Multi Objective GA and simple applications of optimization for engineering systems.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Arora, J.S, Introduction to Optimum Design, McGraw Hill International Ed., NY, 1989.
2. Deb, K. Optimization for Engineering Design: Algorithms and Examples, 2nd Ed., PHI, 1995
3. Rao, S.S. Engineering Optimization: Theory and Practice, New Age International (P) Ltd., 2001
4. Goldberg, D.E Genetic Algorithms in Search and Optimization, Pearson publication, 1990.
5. Koza, J.R. Genetic Programming, MIT Press, 1993
6. Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, 2001.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Design of systems for interdisciplinary engineering applications using suitable optimization technique.

CO2: Apply numerical methods for optimal power flow solutions.
CO3: Optimize the parameters for desired steady state or transient response.
CO4: Optimize the genetic programming parameters.
CO5: Design of multi objective genetic algorithm.

19CSPE308 DATA VISUALIZATION TECHNIQUES SEMESTER III

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- To learn the stages of data visualization.
- To gain knowledge about the methodologies used to visualize large data sets.
- To analyze the process involved in data visualization.
- To identify the principles of interactive data visualization.

- To apply the security aspects involved in data visualization.

UNIT-I: INTRODUCTION **9**

Context of data visualization, Definition, Methodology, Visualization Design Objectives, Key Factors, Purpose, Visualization Function and Tone, Visualization Design Options, Data Representation, Data Presentation, Seven Stages of Data Visualization, Widgets, Data Visualization Tools.

UNIT-II: VISUALIZING DATA METHODS **9**

Mapping, Time Series, Connections and Correlations, Scatterplot Maps, Trees, Hierarchies and Recursion, Networks and Graphs, Info Graphics.

UNIT-III: VISUALIZING DATA PROCESS **9**

Acquiring Data, Tools, Locating and Listing Files Asynchronous Image Downloads, Advanced Web Techniques, using a Database, parsing data, Tools for Gathering Clues, Text Markup Languages, Regular Expressions (regexps), Grammars and BNF Notation, Compressed Data, Vectors and Geometry, Binary Data Formats, Advanced Detective Work.

UNIT-IV: INTERACTIVE DATA VISUALIZATION **9**

Drawing with data, Scales, Axes, Updates, Transition and Motion, Interactivity, Layouts, Geomapping, Exporting, Frameworks.

UNIT-V: SECURITY DATA VISUALIZATION **9**

Port Scan Visualization, Vulnerability Assessment and Exploitation, Firewall Log Visualization, Intrusion Detection Log Visualization, Attacking and Defending Visualization Systems, Creating Security Visualization System.

CONTACT PERIODS:
Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Scott Murray, “Interactive data visualization for the web”, O‟Reilly Media, Inc.,2013.
2. Ben Fry, “Visualizing Data”, O‟Reilly Media, Inc., 2007.
3. Greg Conti, “Security Data Visualization: Graphical Techniques for Network Analysis”, No Starch Press Inc, 2007.
4. Cole Nussbaumer Knaflic, “Storytelling with Data – A Data Visualization guide for Business Professionals”, Wiley Publishers, 2015.
5. Edward R.Tuftle, “The Visual Display of Quantitative Information”, Graphic Press.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the stages of data visualization.
CO2: Design and use various methodologies present in data visualization.
CO3: Discuss the process involved in data visualization.
CO4: Describe the concepts of interactive data visualization.
CO5: Solve the security issues in data visualization for real time applications.

19CSPE309	**COMPUTER GRAPHICS AND MULTIMEDIA**	**SEMESTER III**

L	**T**	**P**	**C**
3	**0**	**0**	**3**

COURSE OBJECTIVES:

- Impart the fundamental concepts of computer graphics.
- Gain knowledge about two dimensional concepts and their applications.
- Learn about three dimensional objects using suitable transformations.
- Learn about the basic concepts of multimedia.
- Apply the knowledge of multimedia for developing applications.

UNIT-I: 2D PRIMITIVES **9**

Computer Graphics Hardware, Software, Introduction to OpenGL, Basic OpenGL Programming, Graphics Output Primitives, Attributes of Graphics Primitives, Implementation Algorithms for Graphics Primitives and Attributes, Line, Circle and Ellipse Drawing Algorithms.

UNIT-II: 2D GEOMETRIC TRANSFORMATIONS AND VIEWING 9

Basic 2D Geometric Transformations, Matrix Representations, Composite Transformations, Reflection and Shearing Transformations, 2D Viewing, Window to View Port Coordinate Transformation, Two Dimensional Viewing Functions, Clipping Operations, Point Clipping, Cohen-Sutherland Line Clipping, Sutherland Hodgman Polygon Clipping.

UNIT-III: 3D CONCEPTS 9

Three Dimensional Object Representation, Polygons, Curved Lines, Splines, Quadric Surfaces, 3D affine Transformations, Parallel and Perspective Projections, Visualization of Data Sets, Viewing, Visible Surface Identification, Color Models, Case Study.

UNIT-IV: MULTIMEDIA BASICS AND 3D MODELLING 9

Introduction and Definitions, Applications, Elements, Animations, Definition of Modelling, Surface Modelling, Object Cloning, Object Editing, 3D Procedural Modelling, Modelling with Polygons, Building Simple Scenes, Building Complex Scenes, Modelling with NURBS.

UNIT-V: MULTIMEDIA APPLICATION DESIGN 9

Types of Multimedia Systems, Virtual Reality Design, Components of Multimedia System, Distributed Application Design Issues, Multimedia Authoring and User Interface, Hypermedia Messaging, Distributed Multimedia Systems.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Donald D. Hearn, M. Pauline Baker and Warren Carithers, “Computer Graphics with OpenGL”, Fourth Edition, Pearson Education, 2014.
2. John F. Hughes, Andries Van Dam, Morgan Mc Guire ,David F. Sklar , James D. Foley, Steven K. Feiner and Kurt Akeley, ”Computer Graphics: Principles and Practice”, Third Edition, Addison, Wesley Professional, 2013.
3. Hill F.S.a and Stephen M.Kelley, “Computer Graphics using OPENGL”, Third Edition, Pearson Education, 2014.
4. Judith Jeffcoate, “Multimedia in practice technology and Applications”, PHI, 1998.
5. Prabat K Andleigh and Kiran Thakrar, “Multimedia Systems and Design”, PHI, 2003.
6. Ze-Nian Li and Mark S. Drew, “Fundamentals of Multimedia”, First Edition, Pearson Education, 2004.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain computer graphics hardware & software and the concepts of 2D primitives.
CO2: Illustrate the various 2D geometric transformations and viewing concepts.
CO3: Explain the various 3D object representations.
CO4: Apply the techniques of multimedia, compression, communication and authoring.
CO5: Design a simple application with animation.

OPEN ELECTIVES I

19CSOE06 COMPONENT BASED SYSTEM DESIGN SEMESTER III

L	T	P	C
3	0	0	3

COURSE OBJECTIVES:

- Describe the importance and need for effective user friendly Graphical User Interfaces(GUI).
- Choose suitable interactions devices/tools to meet application specific requirements.
- Design Graphical User Interfaces(GUI) using apt components and apply the design guidelines for user-friendly navigation and presentation.
- Asses graphical user interfaces for compliance against the screen design guidelines.

UNIT-I: INTRODUCTION 9

Importance of User Interface: Definition – Importance of good design – Benefits of good design – Human-centered development and Evaluation – Human Performance models – A Brief history of screen design.

UNIT-II: THE GRAPHICAL USER INTERFACE & DESIGN PROCESS 9

GUI: Popularity of graphics – The concept of direct manipulation – Graphical system – Characteristics – Web user – Interface Popularity – Characteristics and Principles of User Interface. Design process: Human Interaction with computers – Importance of Human Characteristics – Human Consideration – Human Interaction Speeds and Understanding Business Junctions.

UNIT-III: SCREEN DESIGNING 9

Design Goals – Screen Planning and Purpose – Organizing Screen Elements – Ordering of Screen Data and Content – Screen Navigation and Flow – Visually Pleasing Composition – Amount of Information – Focus and Emphasis – Presenting Information Simply and Meaningfully – Information retrieval on web – Statistical Analysis – Technological considerations in Interface Design.

UNIT-IV: WINDOWS & COMPONENTS 9

Windows: New Navigation Schemes – Selection of Window – Selection of Devices Based on Screen Based Controls. Components: Text and Messages – Icons and Increases – Multimedia – Colors – Uses – Problems – Choosing colors.

UNIT-V: SOFTWARE TOOLS AND INTERACTION DEVICES 9

Specification Methods – Interface Building Tools – Keyboard and Function Keys – Pointing Devices Speech Recognition.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Wilbert O Galitz, "The Essential Guide to User Interface Design", Third Edition, Wiley India Pvt., Ltd., 2007.
2. Ben Shneidermann, "Designing the User Interface", Fifth edition, Pearson Education Asia, 2013. (Software Tools and Interaction Devices)

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Design effective dialog for HCI

CO2: Design effective HCI for individuals and persons with disabilities.

CO3: Assess the importance of user feedback.

CO4: Explain the HCI implications for designing multimedia/ ecommerce/ e-learning Web sites

CO5: Develop meaningful user interface.

19CSOE07	PATTERN RECOGNITION	SEMESTER III			
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

COURSE OBJECTIVES:

- To get knowledge in pattern recognition in computer vision techniques
- To get knowledge in structural pattern methods
- To get knowledge on neural networks and fuzzy systems

UNIT-I: PATTERN CLASSIFIER 9

Overview of pattern recognition – Discriminant functions – Supervised learning –Parametric estimation – Maximum likelihood estimation – Bayesian parameter estimation – Perceptron algorithm – LMSE algorithm – Problems with Bayes approach – Pattern classification by distance functions – Minimum distance pattern classifier.

UNIT-II: UNSUPERVISED CLASSIFICATION 9

Clustering for unsupervised learning and classification – Clustering concept – C-means algorithm – Hierarchical clustering procedures – Graph theoretic approach to pattern clustering – Validity of clustering solutions.

UNIT-III: STRUCTURAL PATTERN RECOGNITION 9

Elements of formal Grammars – String generation as pattern description – recognition of syntactic description – Parsing-Stochastic grammars and applications – Graph based structural representation.

UNIT-IV: FEATURE EXTRACTION AND SELECTION 9

Entropy minimization – Karhunen – Loeve transformation-feature selection through functions approximation – Binary feature selection.

UNIT-V: NEURAL NETWORKS 9

Neural network structures for Pattern Recognition – Neural network based Pattern associators – Unsupervised learning in neural Pattern Recognition – Self organizing networks – Fuzzy logic-Fuzzy classifiers – Pattern classification using Genetic Algorithms.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Duda R. O, P.E Hart and Stork, "Pattern Classification", Wiley, 2012.
2. Robert J. Sehalkoff, "Pattern Recognition: Statistical, Structural and Neural Approaches", JohnWiley & Sons Inc., 2007.
3. Tou & Gonzales, "Pattern Recognition Principles", Wesley Publication Company, 2000.
4. Morton Nadier and P. Eric Smith, "Pattern Recognition Engineering", John Wiley & Sons, 2000.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Apply parametric estimation and supervised learning techniques for pattern classification

CO2: Describe the structural pattern recognition methods

CO3: Apply neural networks, fuzzy systems and Genetic algorithms to pattern recognition and classification

19CSOE08	ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	SEMESTER III			
		L	**T**	**P**	**C**
		3	**0**	**0**	**3**

COURSE OBJECTIVES:

- Artificial Intelligence and intelligent agents, history of Artificial Intelligence
- Building intelligent agents (search, games, constraint satisfaction problems)
- Machine Learning algorithms
- Applications of AI (Natural Language Processing, Robotics/Vision)
- Solving real AI problems through programming with Python, Tensor Flow and Keras library

UNIT-I: FOUNDATIONS OF AI **9**

Introduction – History of Artificial Intelligence – Intelligent Agents – Uninformed Search Strategies – Informed (Heuristic) Search Strategies – Adversarial Search – Constraint Satisfaction Problems.

UNIT-II: SUPERVISED AND UNSUPERVISED LEARNING **9**

Maximum likelihood estimation – Regression – Linear, Multiple, Logistic – bias-variance, Bayes rule, maximum a posteriori inference – Classification techniques – k-NN, naïve Bayes –

Decision Trees – Clustering – k-means, hierarchical, high-dimensional – Expectation Maximization.

UNIT-III: ENSEMBLE TECHNIQUES AND REINFORCEMENT LEARNING 9

Graphical Models – Directed and Undirected Models – Inference – Learning- maximum margin, support vector machines – Boosting and Bagging – Random Forests – PCA and variations – Markov models, hidden Markov models – Reinforcement Learning- introduction – Markov Decision Processes – Value-based methods – Q-learning – Policy-based methods.

UNIT-IV: DEEP LEARNING 9

Neural Network Basics – Deep Neural Networks – Recurrent Neural Networks (RNN) – Deep Learning applied to Images using CNN – Tensor Flow for Neural Networks & Deep Learning.

UNIT-V: AI APPLICATIONS 9

Applications in Computer Vision: Object Detection- Face Recognition – Action and Activity Recognition – Human Pose Estimation. **Natural Language Processing:** Statistical NLP and text similarity – Syntax and Parsing techniques – Text Summarization Techniques – Semantics and Generation – Application in NLP – Text Classification – speech Recognition – Machine Translation – Document Summarization – Question Answering. **Applications in Robotics:** Imitation Learning – Self-Supervised Learning – Assistive and Medical Technologies – Multi-Agent Learning.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Peter Norvig and Stuart J. Russell, “Artificial Intelligence: A Modern Approach”, Third edition
2. Tom Mitchell, “Machine Learning”, McGraw-Hill, 1997
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courvillem, "Deep Learning", MIT press, 2016.
4. Michael Nielson , “Neural Networks and Deep Learning”
5. Christopher Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006
6. Richard Sutton and Andrew Barto, Reinforcement Learning: An introduction”, MIT Press, 1998

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Develop expertise in popular AI & ML technologies and problem-solving methodologies.

CO2: Use fundamental machine learning techniques, such as regression, clustering, knearest neighbor methods, etc.

CO3: Distinguish between supervised and unsupervised machine learning methods.

CO4: Gain knowledge of the different modalities of Deep learning currently used.

CO5: Use popular AI & ML technologies like Python, Tensor flow and Keras to develop Applications

19CSOE09	COMPUTER NETWORK ENGINEERING	SEMESTER III			
		L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- The hardware and software architecture of Computer Networks
- The concepts of internetworking
- Issues in resource allocation
- End-to-end protocols and data transmission
- Network management models

UNIT-I: FOUNDATION **9**

Applications – Requirements – Network Architecture – Implementing Network software – Performance – Perspectives on connecting – Encoding – Framing – Error detection – Reliable transmission – Ethernet and Multiple Access Networks – Wireless.

UNIT-II: INTERNETWORKING **9**

Switching and bridging – IP – Routing – Implementation and Performance – Advanced Internetworking – The Global Internet – Multicast – Multiprotocol and Label Switching – Routing among Mobile devices.

UNIT-III: CONGESTION CONTROL AND RESOURCE ALLOCATION **9**

Issues in Resource allocation – Queuing disciplines – Congestion Control – Congestion avoidance mechanism – Quality of Service.

UNIT-IV: END-TO-END PROTCOLS AND DATA **9**

Simple Demultiplexer – Reliable Byte Stream –Remote Procedure Call – RTP – Presentation formatting - Multimedia data.

UNIT-V: NETWORK MANAGEMENT **9**

SNMPv1 and v2 Organization and information model – Communication model – Functional model – SNMP proxy server – Remote monitoring- RMON1 andRMON2.

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Larry L. Peterson, Bruce S. Davie, "Computer Networks a Systems approach", Fifth edition, Elsevier, 2011.
2. Priscilla Oppenheimer, "Top-down Network Design: A Systems Analysis Approach to Enterprise Network Design", 3rd Edition, Cisco Press, 2010.
3. James D. McCabe, Morgan Kaufmann, "Network Analysis, Architecture, and Design", Third Edition, Elsevier, 2007.
4. William Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," Third Edition, Pearson Education, 2012
5. Mani Subramanian, "Network Management Principles and practice", Pearson Education, 2010.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: Explain the architecture and applications of Computer Networks and analyze the performance of MAC protocols.

CO2: Configure switches and Routers.

CO3: Design algorithms to ensure congestion control and QOS.

CO4: Appreciate the performance of End-to-End protocols and data transmission techniques.

CO5: Use SNMP and RMON.

19CSOE10	GREEN COMPUTING	SEMESTER III			
		L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- To acquire knowledge to adopt green computing practices
- To minimize negative impacts on the environment
- To learn about energy saving practices
- To get knowledge in the impact of e-waste and carbon waste.

UNIT-I: FUNDAMENTALS **9**

Green IT Fundamentals: Business, IT and the Environment – Benefits of a Green Data Centre – Green Computing: Carbon Foot Print, Scoop on Power – Green IT Strategies: Drivers, Dimensions and Goals – Environmentally Responsible Business: Policies, Practices and Metrics.

UNIT-II: GREEN ASSETS AND MODELING **9**

Green Assets: Buildings, Data Centers, Networks, Devices, Computer and Earth Friendly peripherals, Greening Mobile devices – Green Business Process Management: Modeling, Optimization and Collaboration – Green Enterprise Architecture – Environmental Intelligence – Green Supply Chains – Green Information Systems: Design and Development Models.

UNIT-III: GRID FRAMEWORK **9**

Virtualizing of IT Systems – Role of Electric Utilities, Telecommuting, Teleconferencing and Teleporting – Materials Recycling – Best Ways for Green PC – Green Data Center – Green Grid Framework. Optimizing Computer Power Management, Seamless Sharing Across Systems. Collaborating and Cloud Computing, Virtual Presence.

UNIT-IV: GREEN COMPLIANCE **9**

Socio-Cultural Aspects of Green IT – Green Enterprise Transformation Roadmap – Green Compliance: Protocols, Standards, and Audits – Emergent Carbon Issues: Technologies and Future. Best Ways to Make Computer Greener.

UNIT-V: GREEN INITIATIVES WITH IT and CASE STUDIES 9

Green Initiative Drivers and Benefits with IT – Resources and Offerings to Assist Green Initiatives. Green Initiative Strategy with IT – Green Initiative Planning with IT – Green Initiative Implementation with IT – Green Initiative Assessment with IT. The Environmentally Responsible Business Strategies (ERBS) – Case Study Scenarios for Trial Runs – Case Studies – Applying Green IT Strategies and Applications to a Home, Hospital, Packaging Industry and Telecom Sector

CONTACT PERIODS:

Lecture: 45 Periods Tutorial: 0 Periods Practical: 0 Periods Total: 45 Periods

REFERENCES:

1. Bhuvan Unhelkar, “Green IT Strategies and Applications-Using Environmental Intelligence” , CRC Press, June 2011
2. Carl Speshocky, “Empowering Green Initiatives with IT”, John Wiley and Sons, 2010.
3. Alin Gales, Michael Schaefer and Mike Ebbers, “Green Data Center: Steps for the Journey”, Shoff/IBM rebook, 2011.
4. John Lamb, “The Greening of IT”, Pearson Education, 2009.
5. Jason Harris, “Green Computing and Green IT- Best Practices on Regulations and Industry”, Lulu.com, 2008.
6. Woody Leonhard, Katherrine Murray, “Green Home computing for dummies”, August 2009.

COURSE OUTCOMES:

Upon successful completion of the course, students should be able to:

CO1: To explain the necessity of Green IT

CO2: To outline methodologies for creating Green Assets and their management

CO3: To appreciate the use of Grid in Green IT

CO4: To develop case studies related to Environmentally Responsible Business Strategies