College of Engineering, Pune

(An Autonomous Institute of Govt. of Maharashtra, Permanently Affiliated to S.P. Pune University)

Department of Computer Engineering and Information Technology

Curriculum Structure & Detailed Syllabus (PG Program)

M.Tech. – Computer Engineering

(Effective from: A.Y. 2019-20)

Program Educational Objectives (PEOs)

1. To make students eligible to take up higher studies/research
2. To build competency among students to take up jobs that require technical expertise and problem solving ability
3. To inculcate readiness among students for self learning
4. To build competency among students in applying technology to solve real life socio-economic problems

Program Outcomes (POs)

The post-graduate students will demonstrate:

a. In depth knowledge of fundamentals of Computer Engineering and advanced learning in specialization(s)
b. Ability to analyze a problem critically using scientific methodology
c. Ability to explore alternative solutions to solve a problem and decide the best feasible solution under given constraints (time, resources, technology, finance etc) using appropriate modern tools and technology
d. Ability to manage/work in teams with diverse backgrounds in different aspects (such as language, region, technical proficiency, engineering discipline etc) and communicate effectively
e. Ability of life-long self learning and to keep oneself up-to-date in the field of technology
f. Understand intellectual property rights and ability to apply them in an appropriate manner
g. Ability to do SWOT analysis for one-self

Correlation between the PEOs and the POs

PO / PEO	a	b	c	d	e	f	g
1	✓	✓	✓		✓	✓	
2	✓	✓	✓	✓	✓		✓
3		✓	✓		✓		✓
4	✓	✓	✓	✓		✓	✓

Note: The cells filled in wi ✓ indicate the fulfilment/correlation of the concerned PEO with the PO.

List of Abbreviations

Abbreviation	Title	No of courses	Credits	% of Credits
PSMC	Program Specific Mathematics Course	1	4	5.9%
PSBC	Program Specific Bridge Course	1	3	4.4%
DEC	Department Elective Course	3	9	13.2%
MLC	Mandatory Learning Course	2	0	0%
PCC	Program Core Course	6	22	32.4%
LC	Laboratory Course	2	2	2.9%
IOC	Interdisciplinary Open Course	1	3	4.4%
LLC	Liberal Learning Course	1	1	1.5%
SLC	Self Learning Course	2	6	8.8%
SBC	Skill Based Course	2	18	26.5%

Semester I

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1.	PSMC	COC-19001	Probability, Statistics and Queuing Theory	3	1	--	4
2.	PSBC	COC-19002	Algorithms and Complexity Theory	3	--	--	3
3.	DEC		**Department Elective –I**	3	--	--	3
		COC(DE)-19001	1. Distributed Operating System				
		COC(DE)-19002	2. Artificial Intelligence				
		COC(DE)-19003	3. Advanced Graph Theory				
4.	MLC	ML-19011	Research Methodology and Intellectual Property Rights	2	--	--	--
5.	MLC	ML-19012	Effective Technical Communication	1	--	--	--
6.1	PCC	COC-19003	Topics in Databases	3	--	--	3
6.2	PCC	COC-19004	Advanced Computer Networks	3	--	--	3
6.3	PCC	COC-19005	Advanced Computer Architecture	3	--	--	3
6.4	LC	COC-19006	Topics in Databases - Laboratory	--	--	2	1
6.5	LC	COC-19007	Advanced Computer Networks – Laboratory	--	--	2	1
6.6	LC	COC-19008	Advanced Computer Architecture – Laboratory	--	--	2	1
			Total	**21**	**1**	**6**	**22**

Interdisciplinary Open Course (IOC): Every department shall offer one IOC course (in Engineering/Science/Technology). A student can opt for an IOC course offered by a department except the one offered by his /her department.

Semester II

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1.	IOC		*Interdisciplinary Open Course	3	--	--	3
2.	DEC		**Department Elective –II**	3	--	--	3
		COC(DE)-19004	1. Data Analytics				
		COC(DE)-19005	2. Cloud Computing and Virtualization				
		COC(DE)-19006	3. Natural Language Processing				
		COC(DE)-19007	4. Advanced Algorithms				
3.	DEC		**Department Elective –III**	3	--	--	3
		COC(DE)-19009	1. Bioinformatics (BI)				
		COC(DE)-19010	2. Advanced Compiler Construction (ACC)				
		COC(DE)-19011	3. Deep Learning				
		COC(DE)-19012	4. Multicore Technology				
4.	LLC	LL-19001	Liberal Learning Course	1	--	--	1
5.1	PCC	COC-19009	Data Mining and Machine Learning	3		--	3
5.2	PCC	COC-19010	Security in Computing	3		--	3
5.3	PCC	COC-19011	Embedded Systems	3		--	3
5.4	LC	COC-19012	Data Mining and Machine Learning - Laboratory	--		2	1
5.5	LC	COC-19013	Security in Computing - Laboratory	--		2	1
5.6	LC	COC-19014	Embedded Systems - Laboratory	--		2	1
			Total	**19**	**--**	**6**	**22**

***:** Department is going to offer **'Data Structures'** as IOC for students of other departments.

Semester-III

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1.	SBC	COC-19015	Dissertation Phase – I	--	--	12	6
2.	SLC	COC-19016	Massive Open Online Course –I	3	--	--	3
3.	SLC	COC-19017	Massive Open Online Course –II	3	--	--	3
			Total	6	--	12	12

Semester-IV

Sr. No.	Course Type	Course Code	Course Name	Teaching Scheme			Credits
				L	T	P	
1.	SBC	COC-19018	Dissertation Phase – II	--	--	24	12
			Total	--	--	24	12

[COC-19001] Probability, Statistics and Queuing Theory

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
Tutorial: 1hr/week	End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Solve problems related to basic probability theory
2. Solve problems related to basic concepts and commonly used techniques of statistics
3. Model a given scenario using continuous and discrete distributions appropriately and estimate the required probability of a set of events
4. Apply theory of probability and statistics to solve problems in domains such as machine learning, data mining, computer networks etc.

Unit 1: Basic Probability Theory [02 Hrs]

Probability axioms, conditional probability, independence of events, Bayes' rule, Bernoulli trials

Unit 2: Random Variables and Expectation [10 Hrs]

Discrete random variables: Random variables and their event spaces, Probability Mass Function, Discrete Distributions such as Binomial, Poisson, Geometric etc., Indicator random variables, Continuous random variables: Distributions such as Exponential, Erlang, Gamma, Normal etc., Functions of a random variable Expectation: Moments, Expectation based on multiple random variables, Transform methods, Moments and Transforms of some distributions such as Binomial, Geometric, Poisson, Gamma, Normal

Unit 3: Stochastic Processes [6 Hrs]

Introduction and classification of stochastic processes, Bernoulli process, Poisson process, Renewal processes

Unit 4: Markov chains [8 Hrs]

Discrete-Time Markov chains: computation of n-step transition probabilities, state classification and limiting probabilities, distribution of time between time changes, M/G/1 queuing system

Continuous-Time Markov chains: Birth-Death process (M/M/1 and M/M/m queues), Non-birth-death processes, Petri nets

Unit 5: Statistical Inference [8 Hrs]

Parameter Estimation – sampling from normal distribution, exponential distribution, estimation related to Markov chains, Hypothesis testing.

Unit 6: Regression and Analysis of Variance [6 Hrs]

Least square curve fitting, Linear and non-linear regression, Analysis of variance.

Text Book

1. Ronald Walpole, Probability and Statistics for Engineers and Scientists, Pearson, ISBN-13: 978-0321629111

Reference Book

1. Kishor Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, John Wiley and Sons, New York, 2001, ISBN number 0-471-33341-7

[COC-19002] Algorithms and Complexity Theory

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
	End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Determine different time complexities of a given algorithm
2. Demonstrate various design techniques using typical algorithms
3. Develop algorithms using various design techniques for a given problem.
4. Formalize and abstract from a given computational task relevant computational problems, reduce problems and argue about complexity classes

Unit 1: Mathematical Foundation [6 Hrs]

Growth of functions – Asymptotic notation, Standard notation and common functions, Summations, solving recurrences.

Unit 2: Analysis of Algorithms [8 Hrs]

Necessity of time and space requirement analysis of algorithms, Worst case analysis of common algorithms and operations on elementary data structures (e.g. Heapsort), Average case analysis of Quicksort, Amortized analysis.

Unit 3: Standard Design Techniques-I [6 Hrs]

Divide and Conquer, Greedy method.

Unit 4: Standard Design Techniques-II [8 Hrs]

Dynamic programming, Graphs and Traversals.

Unit 5: Standard Design Techniques-III [6 Hrs]

Backtracking, Branch-and-bound.

Unit 6: Complexity Theory [6 Hrs]

Lower-bound arguments, Introduction to NP-Completeness, Reducibility (SAT, 3VC, Independent Set, Subset Sum, Hamiltonian Circuit etc), Introduction to approximation algorithms.

Text Book

1. Thomas Cormen et al., “Introduction to Algorithms” , PHI

Reference Book

1. E. Horowitz and S. Sahni. “Fundamentals of Computer Algorithms” , Galgotia, 1991

[COC(DE)-19001] Distributed Operating Systems

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:

1. Explain Characteristics and challenges of distributed systems.
2. Distinguish the design issues in distributed operating systems
3. Discuss RPC mechanisms, Distributed Shared Memory for inter process communication
4. Summarize the Synchronization issues in Resource and Process management, Process Migration, Distributed File System etc
5. Deconstruct the key components in distributed systems

Unit 1: Fundamentals and Message Passing [10 Hrs]
Fundamentals: Characteristics and challenges of distributed systems. Design issues in distributed operating systems; Architectural models, DCE. Message passing: Desirable features of good message passing systems, Issues in IPC by message passing; Synchronization, Buffering, Multi-datagram Messages, Encoding and decoding of message data, process Addressing, Failure Handling, Group Communication.

Unit 2: Remote procedure Call [7 Hrs]
RPC Model, Transparency of RPC, Implementing RPC mechanisms, RPC messages, Server management, parameter-passing semantics, call semantics Communication protocols for RPC, Client-Server Binding, RPC in Heterogeneous Environment.

Unit 3: Distributed Shared Memory & Synchronization [7 Hrs]
General Architecture of DSM Systems, Design and Implementation issues in DSM, Consistency Models, Implementing Sequential Consistency Model, Page based distributed shared memory, shared – variable distributed shared memory, object-based distributed shared memory. Replacement Strategy, Thrashing, Heterogeneous DSM, Advantages of DSM, Synchronization: Clock Synchronization, Event Ordering, Mutual Exclusion, Deadlock, Election Algorithms.

Unit 4: Resource and Process management [6 Hrs]
Desirable features of good global scheduling algorithms, Task Assignment Approach, Load-Balancing Approach, Load-Sharing Approach, Process management: Process Migration, Threads.

Unit 5: Distributed File System and Naming **[6 Hrs]**
File-Accessing Models, File-Sharing Semantics, File-caching Schemes, File Replication, Fault Tolerance, Atomic Transactions, Design Principles, Naming: Fundamental Terminologies and Concepts, System-Oriented names, Object-Locating Mechanisms, Human-Oriented names, Name cache, Naming and Security.

Unit 6: Security **[6 Hrs]**
Potential Attacks to Computer Systems, Cryptography, Authentication, Access Control, Digital Signatures.

Text Book

1. Sinha P. K., Distributed Operating Systems Concepts and Design, PHI, 1997

Reference Book

1. Tanenbaum A. S., Distributed Operating Systems, Pearson Education India, 1995

[COC(DE)-19002] Artificial Intelligence

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:

1. Apply basic search techniques for problem solving.
2. Explain how to represent Knowledge required for problem solving.
3. Apply reasoning to sift through data.
4. Utilize AI for application in real world.

Unit 1: **[6 Hrs]**
Introduction: Artificial Intelligence, AI Problems, AI Techniques, The Level of the Model, Criteria For Success. Defining the Problem as a State Space Search, Problem Characteristics, Production Systems, Search: Issues in The Design of Search Programs, Un-Informed Search, BFS, DFS; Heuristic Search Techniques: Generate-And- Test, Hill Climbing, Best-First Search, A* Algorithm, Problem Reduction, AO*Algorithm, Constraint Satisfaction, Means-Ends Analysis

Unit 2: **[6 Hrs]**
Knowledge Representation: Procedural Vs Declarative Knowledge, Representations & Approaches to Knowledge Representation, Forward Vs Backward Reasoning, Matching Techniques, Partial Matching, Fuzzy Matching Algorithms and RETE Matching Algorithms

Unit 3: **[6 Hrs]**
Symbolic Logic: Propositional Logic, First Order Predicate Logic: Representing Instance and is-a Relationships, Computable Functions and Predicates, Syntax & Semantics of FOPL, Normal Forms, Unification &Resolution, Representation Using Rules, Natural

Deduction; Structured Representations of Knowledge: Semantic Nets, Partitioned Semantic Nets, Frames, Conceptual Dependency, Conceptual Graphs, Scripts, CYC

Unit 4: **[6 Hrs]**
Reasoning under Uncertainty: Introduction to Non-Monotonic Reasoning, Truth Maintenance Systems, Logics for Non-Monotonic Reasoning, Model and Temporal Logics; Statistical Reasoning: Bayes Theorem, Certainty Factors and Rule-Based Systems, Bayesian Probabilistic Inference, Bayesian Networks, Dempster-Shafer Theory, Fuzzy Logic: Crisp Sets, Fuzzy Sets, Fuzzy Logic Control, Fuzzy Inferences & Fuzzy Systems

Unit 5: **[6 Hrs]**
Natural Language Processing: Role of Knowledge in Language Understanding, Approaches Natural Language Understanding, steps in The Natural Language Processing, Syntactic Processing and Augmented Transition Nets, Semantic Analysis, NLP Understanding Systems; Planning: Components of a Planning System, Goal Stack Planning, Hierarchical Planning, Reactive Systems

Unit 6: **[6 Hrs]**
Machine Learning: Knowledge and Learning, learning by Advise, Examples, learning in problem Solving, Symbol Based Learning, Explanation Based Learning, Version Space, ID3 Decision Based Induction Algorithm, Unsupervised Learning, Reinforcement Learning, Supervised Learning: Perceptron Learning, Back propagation Learning, Competitive Learning, Hebbian Learning.

Text Books
1. Artificial Intelligence, George F Luger, Pearson Education Publications
2. Artificial Intelligence, Elaine Rich and Knight, Mcgraw-Hill Publications

Reference Books
1. Introduction To Artificial Intelligence & Expert Systems, Patterson, PHI 2.
2. Multi Agent systems- a modern approach to Distributed Artificial intelligence, Weiss.G, MIT Press.
3. Artificial Intelligence: A modern Approach, Russell and Norvig, Printice Hall

[COC(DE)-19003] Advanced Graph Theory

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:
1. Prove theorems related to concept of ‘Matching’ in graphs
2. Prove various results related to connectivity in graphs, network flows etc
3. Demonstrate familiarity with the results related to graph coloring
4. Demonstrate familiarity with special classes of graphs, algebraic graph theory and analytic graph theory

Unit 1: Matching and Factors **[8 Hrs]**
Matchings in Bipartite Graphs, Hall's Matching Condition, Min-Max Theorems, Independent Sets, Tutte's 1-Factor Theorem, Maximum Bipartite Matching , Stable Matching, Dominating set andpath cover, Gallai-Millgram theorem. Dilworth's theorem.

Unit 2: Connectivity and Paths **[6 Hrs]**
Cuts and Connectivity, Flows in Directed Graphs, Connectivity: vertex connectivity, edge connectivity, 2-connected and 3-coonected graphs, Menger's theorem and its applications, Network flows: Min cut max flow theorem.

Unit 3: Graph Coloring **[6 Hrs]**
Vertex coloring, Brook's theorem, Edge coloring, Planarity, 5-coloring planar graphs, Kuratowsky's theorem.

Unit 4: Special classes of graphs **[6 Hrs]**
Perfect graphs, Interval Graphs, Chordal Graphs, Weak perfect graph theorem.

Unit 5: Algebraic Graph Theory **[6 Hrs]**
Graphs and matrices, Automorphisms, Cayley Graphs, Spectral Graph Theory.

Unit 6: Analytic Graph Theory **[8 Hrs]**
Extremal graph theory, Random Graphs, Ramsey theory, Probabilistic method.

Text Books

1. Douglas B. West, "Introduction to Graph Theory", Pearson Education India; 2nd edition (2015), ISBN-10: 9789332549654, ISBN-13: 978-9332549654
2. Béla Bollobás, Modern Graph Theory, Springer, 2013, ISBN-10: 9788181283092, ISBN-13: 978-818128309

Reference Books

1. Reinhard Diestel, Graph Theory, 4th edition (2010), ISBN-10: 3642142788, ISBN-13: 978-3642142789
2. Adrian Bondy and U.S.R. Murty, "Graph Theory", Springer, 1st edition (2008), ISBN-10: 1846289696, ISBN-13: 978-1846289699

Internet Resource

1. NPTEL Course: https://nptel.ac.in/courses/106108054/

[COC-19003] Topics in Databases

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:

1. Understand foundation of the RDBMS theory
2. Understand internal functioning of RDBMS
3. Understand advanced topics of RDBMS
4. Analyze and understand the latest trends of RDBMS

Unit 1: Transaction Management [10 hrs]
Overview of transaction management: Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Concurrency Control: Lock-Based Protocols, Timestamp-Based Protocols, Validation-Based Protocols, Recovery System: Recovery and Atomicity, Log-Based Recovery, Recovery with Concurrent Transactions, ARIES (Algorithm for Recovery and Isolation Exploiting Semantics), which supports partial rollbacks of transactions, fine granularity (e. g., record) locking and recovery using write-ahead logging (WAL)

Unit 2: Query Execution [10 hrs]
Architecture of Query Execution Engines, Disk Access, Measures of Query Cost, Introduction to Physical-Query-Plan Operators, One-Pass Algorithms for Database Operations, Nested-Loop Joins, Two-Pass Algorithms Based on Sorting, Two-Pass Algorithms Based on Hashing, Index-Based Algorithms, Algorithms Using More Than Two Passes.

Unit 3: Query Optimization [10 hrs]
Basic Optimization Strategies, Algebraic Manipulation, Optimizations of Selections in System R.

Unit 4: Case Studies [10 hrs]
Hadoop Distributed File System: Study of Hadoop Distributed File System, HIVE - Data warehousing application built on top of Hadoop, MapReduce framework, Dynamo – a structured storage system, Eventual Consistency Model for Distributed Systems.

Text Books and Research Papers

1. Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, “Database System:The Complete Book”, Pearson, 2nd edition (2008), ISBN-10: 0131873253, ISBN-13: 978-0131873254
2. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, McGraw Hill Education, 3rd edition (2014), ISBN-10: 9339213114, ISBN-13: 978-9339213114
3. C. Mohan, “ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging”, ACM Transactions on Database Systems, Vol. 17, No. 1, March, 1992, pp. 94–162.
4. P. Selinger, M. Astrahan, D. Chamberlin, Raymond Lorie and T. Price, “Access Path Selection in a Relational Database Management System”, Proceedings of ACM SIGMOD, pp 23-34, 1979
5. Jeffrey Dean and Sanjay Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Communications of the ACM, vol. 51, no. 1, pp. 107-113, 2008

6. Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, Proceedings of Operating Systems Design and Implementation , pp. 205-218, 2006.
7. W. Vogels, “Eventually Consistent”, ACM Queue, vol. 6, no. 6, December 2008
8. Goetz Graefe, “Query Evaluation Techniques for Large Databases”, ACM Computing Surveys, Vol. 25, No. 2, June 1993

Reference Books

1. Korth, Silberschatz and Sudarshan, “Database System Concepts”, Tata McGraw Hill, 6th edition (2013), ISBN-10: 9332901384, ISBN-13: 978-9332901384
2. R. Elmasri, and S. Navathe, “Fundamentals of Database Systems”, Pearson, 7th edition (2017), ISBN-10: 9789332582705, ISBN-13: 978-9332582705

Internet Resource

1. http://hadoop.apache.org

[COC-19004] Advanced Computer Networks

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Understand issues in the design of network processors and design network systems.
2. Analyze different possible solutions for communications at each network layer.
3. Simulate working of wired and wireless networks to understand networking concepts.
4. Develop solutions by applying knowledge of mathematics, probability, and statistics to network design problems.
5. Understand and compare various storage and networking technologies.

Unit 1 **[4 Hrs]**

Routing Algorithms, Congestion Control, Quality of Service, Queue Management, High Speed Networks, Performance Modeling and Estimation

Unit 2 **[6 Hrs]**

IPv4 deficiencies, patching work done with IPv4, IPv6 addressing, multicast, Anycast, ICMPv6, Neighbor discovery, Routing

Unit 3 **[6 Hrs]**

Centralized and Distributed Control and Data Planes, SDN Controllers, Data Center Concepts, Network Function Virtualization, Mininet, Programming SDNs, Openflow Switch, Wire Protocol, Openstack Neutron plug-in

Unit 4 **[6 Hrs]**

MAC Protocols for Ad Hoc Wireless Networks, Routing Protocols for Ad Hoc Wireless Networks, Multicast routing in Ad Hoc Wireless Networks, Transport Layer and Security Protocols for Ad Hoc Wireless Networks, Quality of Service in Ad Hoc Wireless Networks.

Unit 5 **[6 Hrs]**

SNMPv1 Network Management: Organization and Information Models, SNMPv2: major changes, SNMPv3, RMON, Network Management Tools, Systems, and Engineering, Network Management Applications.

Unit 6 **[6 Hrs]**

Storage and Networking Concepts, Fiber Channel Internals, Fiber Channel SAN Topologies, Fiber Channel Products, IP SAN Technology, IP SAN Products, Management of SANs, SAN Issues.

Text Books

1. Mani Subramanian, Timothy A. Gonsalves,N. Usha Rani; Network Management: Principles and Practice; Pearson Education India, 2010
2. William Stallings, High-Speed Networks and Internets, Pearson Education, 2nd Edition, 2002.
3. C. Siva Ram Murthy, B.S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall, 2004
4. Muthukumaran B, Introduction to High Performance Networks, Tata Mc Graw Hill, 2008

Reference Books

1. Thomas D NAdeau and Ken Grey, Software Defined Networking, O'Reilly, 2013
2. Pete Loshin IPv6, Theory, Protocols and Practice, Morgan Kaufmann, 2nd Edition, 2004
3. Tom Clark, Designing Storage Area Networks, A Practical Reference for Implementing Fibre Channel and IP SANs, Addison-Wesley Professional, 2nd Edition, 2003.

[COC-19005] Advanced Computer Architecture

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
	End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Understand the history , evolution , classifications & current trends of Computer Architecture; Learn to evaluate & compare System’s performance using standard benchmarks
2. Understand the basics of advanced microprocessor techniques & the salient features of state-of-the- art processors deployed in current High Performance Computing systems

3. Understand the differences between System Area Networks & Storage Area Networks & learn the current Networking Technologies for implementing them
4. Learn the advanced RAID Levels, compare SAS vs SATA Disks & understand the implementation of a hierarchical Storage System
5. Understand the System Software Architecture, various parallel programming models, message passing paradigms & typical HPCC software stack
6. Understand, through the case studies of a few selected representative systems, the implementation of architectural concepts learnt through CO-1 to CO-5

Unit 1: System Architecture **[8 Hrs]**
History /Evolution, Definition: Hardware /Software Architecture, Flynn's Classification: SISD,SIMD,MISD,MIMD. Physical Models: PVP, MPP, SMP& Cluster of Workstations (COW). Memory Architectures: Shared, Distributed & Hybrid. UMA, NUMA, CC-NUMA. Performance Metrics & Benchmarks (Micro/Macro) Architectural Trends based on TOP 500 List of Supercomputers.

Unit 2: Advanced Microprocessor Techniques **[8 Hrs]**
CISC, RISC, EPIC, Superscalar, Superpipelined Architectures, Superscalar/ Superpipelined, In Order Execution /Out of Order Execution (OOO), ILP, TLP, Power Wall, Moore's Law Redefined, Multicore Technologies, Intel's Tick-Talk Model. Study of State-of-the- ART Processors : Intel / AMD X86-64 Bit Series: Intel Xeon Family (Xeon Haswell & Broadwell Architectures), Intel Xeon Phi Coprocessors (MIC Architecture) Intel/IBM Itanium/Power Series (Power 4 - Power 9). Introduction to Graphics Processing Units (GPU-NVIDIA).

Unit 3: System Interconnects **[4 Hrs]**
SAN : System Area Networks, Storage Area Networks including InfiniBand, Gigabit Ethernet, Scalable Coherent Interface (SCI) Standard.

Unit 4: Storage **[4 Hrs]**
Internal/External , Disk Storage, Areal Density ,Seek Time ,Disk Power, Advanced RAID Levels, SATA vs SAS Disks ,Network Attached Storage (NAS) ,Direct Attached Storage (DAS), I/O Performance Benchmarks.

Unit 5: Software Architecture **[8 Hrs]**
Parallel Programming Models: Message Passing ,Data Parallel , MPI/PVM .Typical HPCC Software Stack including Cluster Monitoring Tools, Public Domain Software like GANGLIA, CUDA Programming Environment.

Unit 6: Case Studies **[8 Hrs]**
A typical Peta Scale System based on Hybrid CPU/GPU Architectures, IBM SP System, C-DAC's latest PARAM Systems [PARAM Yuva-II], Sequent NUMA Q, Case Study of a Domain Specific Architecture (DSA).

Text Books

1. John L. Hennesy and David Patterson, Computer Architecture : A Quantitative Approach, 6th Edition, Elsevier

2. Kai Hwang and Zhiwei Xu, Scalable Parallel Computers, McGraw- Hill, 1998.
3. Data Manuals of respective Processors available at Website

Reference Book

1. Peter S. Pacheco, “An Introduction to Parallel Programming”, Morgan Kaufmann, Elsevier Series, 2011, ISBN:978-0-12-374260-5.

[COC-19006] Topics in Databases Laboratory

Teaching Scheme
Practical: 2 hrs/week

Examination Scheme
Term Work: 50 marks
Oral Examination: 50 marks

Course Outcomes

Students will be able to:

1. Relate theory with practice by performing programming assignments
2. Analyze various algorithms and implementation options to solve a problem
3. Demonstrate familiarity with how query optimization s done in real life database systems
4. Apply MapReduce technique for solving a few problems in distributed manner

Suggested List of Assignments

1. Write a program to check if a given schedule is serial, serializable, conflictserilaizable.
2. Simulate recovery using undo, redo and undo-redo logging.
3. Simulate ARIES recovery system.
4. Implement disk-based algorithms for sorting.
5. Implement hash-based natural join, sort-based natural-join.
6. Study MySQL/Postgres query optimizer code.
7. Comparing the query evaluation performance before and after applying query optimization techniques.
8. Implementation of MapReduce algorithms for Matrix multiplication and GroupBy operation

[COC-19007] Advanced Computer Networks Laboratory

Teaching Scheme
Practical: 2 hrs/week

Examination Scheme
Term Work: 50 marks
Oral Examination: 50 marks

Course Outcomes

Students will be able to:

1. Relate theory with practice by performing programming assignments
2. Get proficiency in designing programming solutions
3. Get proficiency in variety of tools and environments like C, C++, Java, and Linux OS
4. Analyze various algorithms and implementation options to solve a problem
5. Learn to work in teams while carrying out the assignments
6. Imbibe good programming practices

Suggested List of Assignments

1. Create a Virtual Machine setup and perform socket programming.
2. Implement intradomain routing algorithms.
3. Implement TCP congestion control algorithms.
4. Perform Wireshark packet sniffing experiment.
5. Do the network analysis using NMAP - the Network MAPper.
6. SSH to coep.org.in and run a traceroute to google.com. List the results, then interpret and report your findings.
7. Implement a web proxy that passes requests and data between multiple web clients and web servers.
8. Write an Internet chat server by using Berkeley Sockets API.
9. Design a client server application for solving roots of a quadratic equation by making use of appropriate API's.

[COC-19008] Advanced Computer Architecture Laboratory

Teaching Scheme
Practical: 2 hrs/week

Examination Scheme
Term Work: 50 marks
Oral Examination: 50 marks

Course Outcomes

Students will be able to:

1. Analyze performance of programs using perf tool.
2. Implement shared memory programs using OpenMP.
3. Implement message passing programs in distributed environment.
4. Program GPU architecture using CUDA.
5. Demonstrate the different steps involved in building of a simple cluster.

Suggested List of Assignments

1. Study and installation of 'perf tool' to understand performance details of matrix multiplication program
2. MPI - OpenMP cluster setup over LAN and executing matrix multiply on it
3. CUDA/ OpenACC environment setup and execution of matrix multiply code
4. Study of Intel PIN tool
5. Case study of any architectural open source simulator for any architecture simulation and modelling

Reference Books

1. Peter S. Pacheco, “An Introduction to Parallel Programming”, Morgan Kaufmann, Elsevier Series, 2011, ISBN:978-0-12-374260-5.
2. Jason Sanders, Edward Kandrot, “CUDA by Example: An Introduction to General Purpose GPU Programming”, 2011, ISBN:978-0-13-138768-3.

[IOC] Data Structures

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:
1. Write code by following coding standards, by selecting appropriate data structure and demonstrate a working solution for a given problem
2. Think of all possible inputs to an application and handle all possible errors properly.
3. Analyze clearly different possible solutions to a program and select the most efficient one.
4. Demonstrate the ability to write reusable code and abstract data types, using object-based way of thinking.

Unit 1 **[6 Hrs]**
Introduction: Concept of Data types and Abstract Data types; Characteristics of an algorithm; Analyzing programs; Frequency count; Time and space complexity; Big 'O' and ΈΩ‟ notation; Best, average and worst cases; Programming language provided data types, operations on various data types; Dangling pointers and garbage memory.

Unit 2 **[6 Hrs]**
Arrays, Searching and Sorting: Searching: linear and binary search algorithm; Hashing: hashing functions, chaining, overflow handling with and without chaining, open addressing: linear, quadratic probing; Sorting: bubble sort, selection sort, quick sort, merge sort, insertion sort. Time complexity analysis of searching and sorting techniques.

Unit 3 **[8 Hrs]**
Lists: List as ADT. Concept of linked organization of data against linked list. Singly linked list, doubly linked list, circular linked list. Representation & manipulations of polynomials/sets using linked lists. Dynamic memory management. Representation of sparse matrix. Addition and transpose of sparse matrix.

Unit 4 **[8 Hrs]**
Stacks and Queues: Stack and queue as ADT. Operations on stack and queue. Implementations using arrays and dynamic memory allocation. Application of stack for expression evaluation, expression conversion. Recursion and stacks. Problems like maze and knight's tour.

Unit 5 **[8 Hrs]**
Graph Algorithms: DFS, BFS, Biconnected components, Cut vertices, Matching, Network flow; Maximum-Flow / Minimum-Cut; Ford–Fulkerson algorithm, Augmenting Path.

Unit 6 **[8 Hrs]**

Trees: Basic terminology. Binary trees and its representation. Binary tree traversals (recursive and non recursive) and various operations. Insertion and deletion of nodes in binary search tree.

Text Books

1. "Fundamentals of Data Structures in C", E. Horowitz, S. Sahni, S. Anderson-freed, Second Edition, University Press, ISBN 978-81-7371-605-8
2. "The C Programming Language", B. Kernighan, D. Ritchie, Prentice Hall of India, Second Edition, ISBN 81-203-0596-5

Reference Books

1. "Fundamentals of Data Structures in C++", Ellis Horowitz, S. Sahni, D. Mehta Galgotia Book Source, New Delhi 1995 ISBN 16782928
2. "An introduction to data structures with Applications", Jean-Paul Tremblay, Paul. G. Soresan, Tata Mc-Graw Hill International Editions, 2nd edition 1984, ISBN-0-07-462471-7
3. "Data Structures using C", Y. Langsam, M. Augenstin and A. Tannenbaum, Pearson Education Asia, First Edition, 2002, ISBN 978-81-317-0229-1
4. C++: The Complete Reference, 4th Edition by Herbert Schildt, Fourth Edition, The McGraw-Hill company,ISBN 0-07-222680-3

[COC(DE)-19004] Data Analytics

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
	End-Sem Exam – 60 marks

Course Outcomes

Student will be able to:

1. Identify and assess the opportunities, needs and constraints for data collection, and explore various types of datasets and features.
2. Analyze the business issues that data science and analytics can address and resolve.
3. Identify the methods by which data can be collected, stored, secured, analyzed, interpreted, forecasted, visualized, reported and applied in a business environment
4. Describe how data can be interpreted beyond its basic analysis to tell a story relevant and meaningful to its organization, and how these stories can be utilized to gain competitive advantage through strategic application
5. Design case studies on social media analytics.

Unit 1: Fundamentals of Data Analytics [8 Hrs]

Data Analytics Basics, Data Types, Analytics Types, Data Analytics Steps: Data Pre-Processing, Data Imputation, Data Cleaning, Data Transformation, Data Visualization, and Data Engineering. Descriptive, Predictive, and Prescriptive Analytics.

Unit 2: Data Analytics with Python [8 Hrs]

Data Analytics using Python, Statistical Procedures, Web Scraping in Python, Advanced analytics, NumPy, Pandas, SciPy, Matplotlib.

Unit 3: Correlated Data Analysis [7 Hrs]

Analysis of Variance and Co-Variance, ANOVA results, Chi-Square Statistical Test, Examine Regression results, Regressing Analysis, Linear Regression and its analysis, Logistic Regression and its analysis.

Unit 4: Decision Trees and Cluster Analysis [9 Hrs]

Decision Tree Problem Analysis, Decision tree Construction, Decision Tree Algorithms; Applications of Cluster Analysis, Definition of Cluster, representing clusters, Clustering Techniques, K-Means Algorithm for Clustering, Advantages and Disadvantages of K-Means Clustering.

Unit 5: Social Media Analytics [8 Hrs]

Datasets, Analysis of Social Network Dataset Features, Learning Models and Validation, Association Rule Mining, Artificial Neural Networks for web analytics.

Text Books

1. Anil Maheshwari, “Data Analytics made accessible,” Amazon Digital Publication, 2014.
2. Song, Peter X. -K, “Correlated Data Analysis: Modeling, Analytics, and Applications”, Springer-Verlag New York 2007.
3. Glenn J. Myatt, Wayne P. Johnson, “Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining”, Wiley 2009.

Reference Books

1. Thomas H. Davenport, Jeanne G. Harris and Robert Morison, “Analytics at Work: Smarter Decisions, Better Results”, Harvard Business Press, 2010
2. Rachel Schutt, Cathy O’Neil, “Doing Data Science”, O’REILLY, 2006.
3. Shamanth Kumar Fred Morstatter Huan Liu “Twitter Data Analytics”, Springer-Verlag, 2014.

[COC(DE)-19005] Cloud Computing and Virtualization

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
	End-Sem Exam – 60 marks

Course Outcomes

Student will be able to:

1. Characterize the distinctions between various cloud models and services
2. Compare the functioning and performance of virtualization of CPU, memory and I/O with traditional systems
3. Familiar with cloud platforms and technologies like AWS, vSphere etc.
4. Create a cloud infrastructure after learning OpenStack components

5. Analyze the security risks associated with virtualization, cloud computing and evaluate how to address them

Unit 1 **[6 Hrs]**
Introduction: Benefits and challenges to Cloud architecture, Cloud delivery models- SaaS, PaaS, LaaS. Cloud Deployment Models- Public Cloud, Private Cloud, Community Cloud and Hybrid Cloud, Service level agreements in clouds, case Studies on Cloud services, Cloud Adoption Challenges.

Unit 2 **[10 Hrs]**
Virtualization: Role of virtualization in enabling the cloud, Levels of Virtualizations, Types of Virtualization: Compute, Network and Storage Virtualizations, Virtual Machine, Hypervisor: Type 1 and 2, Examples of Hypervisors.
Server Virtualization: X86 architecture, Protected mode, Rings of Privileges, Virtualization challenges, Full virtualization and Binary Translation, Ring De-Privileging, Handling Sensitive instructions, ESXi, Para-Virtualization, Xen, Hardware Assisted Virtualization, System call and hardware interrupts handling in virtualized systems, Intel VTx, KVM, VM Migration

Unit 3 **[8 hrs]**
Memory and I/O Virtualization: Memory management and I/O with traditional OS, Challenges in virtualized system, Shadow page Tables in Full Virtualized system, EPT/NPT, 2D Page walks, I/O in Virtualized Systems, Emulation, Split drivers of Xen, Direct I/O, Intel VTd, VMCS

Unit 4 **[6 Hrs]**
Cloud Orchestration: Elements of Cloud Orchestration, Examples platforms: OpenStack and vSphere, OpenStack Deep dive: Covers Networking, Storage, Authentication modules of OpenStack, Nova, Quantum, Keystone and Cinder, Swift.

Unit 5 **[4 hrs]**
Cloud Platforms: Overview and Architecture, Azure, Google App Engine, Amazon Web Services.

Unit 6 **[8 Hrs]**
Virtualization Security: Security Challenges Raised by Virtualization, Virtualization Attacks, VM Migration Attacks, Launch Pad for Brute Force attacks, Security Solutions, Hypervisor-Based Segmentation, case studies of Hypervisors.
Cloud Security: Issues with Multi-tenancy, Isolation of users/VMs from each other, VM vulnerabilities, hypervisor vulnerabilities, VM migration attacks, Cloud based DDOS, Developing cloud security models, end-to-end methods for enforcing Security, Security policies and programming models with privacy aware APIs

Text Books

1. Kai Hwang, Geoffrey and KJack, Distributed and Cloud computing, Elsevier
2. Shailendra Singh, Cloud Computing, Oxford Higher Education, , 2018

Reference Books

1. Danielle Ruest and Nelson Ruest, Virtualization, A beginners Guide, Tata McGraw Hill,2009
2. Tom White, Hadoop: The Definitive Guide, O'REILLY, 3rd Edition, 2012
3. Dinakar Sitaram and Geetha Manjunath, Moving to the cloud, Elsevier

On-line Course Resources

1. Understanding Full Virtualization, Para Virtualization and Hardware Assist, VMware White paper
2. AMD-V Nested Paging, white paper, July 2008
3. Patent: US 8533428 B2, Translating a Guest Virtual Address to a Host Physical Address as Guest Software Executes on a Virtual Machine, 2013
4. Darren Abramson, et. all, Intel Virtualization Technology for Directed I/O, Intel Technology Journal, Vol. 10, Issue 3, 2006
5. Uhlig, R., et al., "Intel Virtualization Technology", IEEE Computer Society, 38(5), pp 48-56, , 2005
6. "OpenStack Docs: Current", http://docs.openstack.org/
7. " vSphere 5 Documentation Center: ",http://pubs.vmware.com/vsphere-50/index.jsp
8. "Google App Engine", https://developers.google.com/appengine/
9. "Windowsazure :Microsoft's Cloud Platform| Cloud hosting |Cloud Service ", http://www.windowsazure.com/en-us/
10. Hadoop Performance Tuning, Impetus Technologies Inc., October 2009

[COC(DE)-19006] Natural Language Processing

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
	End-Sem Exam – 60 marks

Course Outcomes

Student will be able to:

1. Demonstrate the understanding of basic text processing techniques in NLP.
2. Design, implement and evaluate part-of-speech taggers and parsers for a language.
3. Build language models and demonstrate Word Sense Disambiguation using WordNet.
4. Analyze and build word embeddings for different languages.

Unit 1: Introduction [6 Hrs]

What is NLP, Fundamental and Scientific goals, Engineering goals, stages of NLP, problems in NLP, Applications of NLP, Empirical Laws of language, zipf's law, Heap's law.

Unit 2: Basic Text Processing [8 Hrs]

Tokenization, word token, word type, sentence segmentation, feature extraction, issues in tokenization for different languages, word segmentation, text segmentation, normalization, case folding, Spelling Correction, Morphology, Stemming, Porters Algorithm, , lemmatization, spelling correction - dynamic programming approach for finding edit distance, N-gram Language Modeling- context sensitive spelling correction, probabilistic language model, auto completion prediction, Evaluation and perplexity, Smoothing techniques.

Unit 3: POS Tagging **[8 Hrs]**
Sequence labeling tasks of NLP, POS tagging, POS tag sets, Hidden Markov Model-Introduction, Markov Processes, HMM characterization -Likelihood of a sequence (Forward Procedure, Backward Procedure), Best state sequence (Viterbi Algorithm), Re-estimation(Baum-Welch - Forward-Backward Algorithm) , Models for Sequential tagging – Maximum Entropy, Conditional Random Field.

Unit 4: Syntax **[10 Hrs]**
Constituency and dependency parsing, Constituency parser -Syntactic structure, Parsing methodology, Different parsing algorithms, Parsing in case of ambiguity, Probabilistic parsing, CKY algorithm, Issues in parsing, Dependency parsing- Syntactic structure, Parsing methodology, Transition-Based Dependency Parsing, Graph-Based dependency parsing, Evaluation, Co-reference resolution, Named-entity recognition.

Unit 5: Knowledge Base and Semantics **[6 Hrs]**
WordNet: Word Senses, Word relations, Word similarity and thesaurus methods, Word sense disambiguation, WordNet. Lexical and Distributional Semantics - Introduction, models of semantics, applications.

Unit 6: Word Embeddings **[6 Hrs]**
Introduction, one-hot vectors, methods of generating word embeddings, Skip-gram, CBOW, Glove model, Fast Text model, evaluation measures-rough scores.

Text Books

1. Daniel Jurafsky and James H. Martin, “Speech and Language Processing”, Second Edition, Prentice Hall, 2008, ISBN: 978-0131873216.
2. Allen James, “Natural Language Understanding”, Second Edition, Benjamin/Cumming, 1994, ISBN: 978-0805303346.
3. Chris Manning and Hinrich Schuetze, “Foundations of Statistical Natural Language Processing”, MIT Press, ISBN: 978-0262133609.

Reference Books

1. Journals: Computational Linguistics, Natural Language Engineering, Machine Learning, Machine Translation, Artificial Intelligence.
2. Conferences: Annual Meeting of the Association of Computational Linguistics (ACL), Computational Linguistics (COLING), European ACL (EACL), Empirical Methods in NLP (EMNLP), Annual Meeting of the Special Interest Group in Information Retrieval (SIGIR), Human Language Technology (HLT).

[COC(DE)-19007] Advanced Algorithms

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each

Tutorial: 1hr/week End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Demonstrate familiarity with various algorithm design paradigms
2. Analyze algorithms based on different paradigms
3. Decide which paradigm suits best to solve a real life problem under given constraints/resources
4. Implement and test representative algorithms based on different techniques

Unit 1: Computational Geometry [8 Hrs]

Convex hull, Closest pair of points, Line segment intersection

Unit 2: Approximation Algorithms [8 Hrs]

Vertex cover problem, Traveling Salesperson problem, Set covering problem

Unit 3: Randomized Algorithms [8 Hrs]

Quick sort, Max-cut algorithm, Primality Testing

Unit 4: Parallel Algorithms [8 Hrs]

Theoretical models of parallel computation: variants of the PRAM model, Performance of parallel algorithms, Quick sort, Bitonic sort

Unit 5: Streaming Algorithms [8 Hrs]

Finding frequent items deterministically, Estimating the number of distinct elements, Finding frequent items via Sketching, Estimating frequency moments

Text Books

1. Thomas Cormen, Charles Leiserson, Ronald Rivest and Cliford Stein, “Introduction to Algorithms” , PHI
2. Jure Leskovec, Anand Rajaraman and Jeffrey Ullman, “Mining of Massive Datasets”, Dreamtech Press

Reference Books

1. Rajeev Motwani and Prabhakar Raghavan, “Randomized Algorithms”, Cambridge University Press
2. Vijay V. Vazirani, "Approximation Algorithms", Springer

[COC(DE)-19009] Bioinformatics

Teaching Scheme

Lectures: 3 hrs/week

Examination Scheme

T1, T2 – 20 marks each

End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Learn various algorithms for sequencing and alignments
2. Implement proof of concepts for the algorithm studied with some sample data
3. Evaluate how networks, algorithms, and models are employed in bioinformatics

4. Compare the molecular biology techniques for drug design for various diseases

Unit 1: Introduction [6 Hrs]

Introduction, chronological history of Bioinformatics, evolution of Bioinformatics, Objectives of Bioinformatics, Importance of bioinformatics, Bioinformatics inbusiness, future scope of Bioinformatics. Protein Structure and Functions, Nature of Chemical Bonds Molecular Biology tools, Polymerase chain reaction.

Unit 2: Sequence Alignment [6 Hrs]

Simple alignments, Gaps, Scoring Matrices, Global and Local Alignments, Smith-Waterman Algorithm, BLAST, FASTA, Multiple sequence Alignments, Gene Prediction, Statistical Approaches to Gene Prediction, Spliced Alignment

Unit 3: Genome Algorithms [8 Hrs]

The dawn of sequencing, the biological sequence or structure deficit, humangenome project and its status, homology and analogy, web browsers. Genome Rearrangements, Sorting by Reversals, Block Alignment and the Four-Russians Speedup, Constructing Alignments in Sub-quadratic Time, Protein Sequencing and Identification, the Peptide Sequencing Problem

Unit 4: Microarray Data Analysis [6 Hrs]

Microarray technology for genome expression study, Image analysis for data extraction, Data analysis for pattern discovery, gene regulatory network analysis. Building a sequence search protocol, practical approach for structural and functional interpretation.

Unit 5: Phylogenetic Analysis [8 Hrs]

Neighbor's relation method, Neighbor-joining method, Maximum likelihood Approaches, Multiple Sequence Methods Structural Biology, Sequence, organisms, 3D structures, complexes, Assemblies, Case Studies, examples.

Unit 6: Drug Discovery &Design [6 Hrs]

Similarities/differences between drugs and receptors, Target identification , Target Validation , Lead Identification ,lead optimization , preclinical Pharmacology & Taxology, protein-ligand docking, Massively Parallel Signature Sequencing (MPSS), SOLiD sequencing, Single molecule real time (SMRT) sequencing.

Text Books

1. Dan E. Krane, Michael L. Raymer, "Fundamental Concepts of Bioinformatics,", Pearson Education, Inc. Fourth Edition, 9780805346336.
2. Harshawardhan P. Bal, "Bioinformatics Principles and Applications", Tata McGraw-Hill, seventh reprint, 9780195692303.

Reference Books

1. Teresa Attwood, David Parry-Smith, "Introduction to Bioinformatics", Pearson Education Series, 9788180301971
2. R. Durbin, S. Eddy, A. Krogh, G. Mitchison., "Biological Sequence Analysis: Probabilistic Models of proteins and nucleic acids", Cambridge University Press, 9780521629713.

3. Arthur M. Lesk, Introduction to Bioinformatics, Oxford University Press, 3rd Edition,2008
4. Andreas D. “Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins”, Second Edition,

[COC(DE)-19010] Advanced Compiler Construction

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:
1. Demonstrate familiarity the design of a typical compiler
2. Design a prototype for the toy compiler
3. Demonstrate familiarity with code optimization techniques
4. Demonstrate familiarity with the issues and techniques related to data flow analysis and register allocation

Unit 1: Introduction [6 Hrs]
Review of Compiler Structure, Advanced Issues in Elementary Topics, Importance of Code Optimization, Structure of Optimizing Compilers, Placement of Optimizations in Aggressive Optimizing Compilers

Unit 2: Context –Sensitive Analysis & Intermediate Representation [6 Hrs]
Introduction to type systems, The Attribute –grammar framework, Adhoc Syntax directed translation, Harder problems in type inference and changing associativity, Issues in designing an intermediate languages, Graphical & Linear IR, Static-single Assignment form, Mapping values to names & symbol tables.

Unit 3: Code Optimization [8 Hrs]
Introduction, Redundant expressions, Scope of optimization, Value numbering over regions larger than basic blocks, Global redundancy elimination, Cloning to increase context, Inline substitution, Introduction to control flow analysis, Approaches to control flow analysis, Interval analysis and control trees, Structural analysis, Reaching definitions.

Unit 4: Data Flow Analysis & Scalar Optimization [10 Hrs]
Basic concepts : Lattices, flow functions and fixed points, Iterative data flow analysis, Lattice of flow functions, Control –tree based data flow analysis, Structural analysis and interval analysis, Static Single Assignment (SSA) form, Dealing with arrays, structures and pointers, Advanced topics: Structures data-flow algorithms and reducibility, Inter procedural analysis (Control flow, data flow, constant propagation, alias), Inter procedural register allocation, Aggregation of global references, Introduction to scalar optimization, Machine –independent and dependent transformations, Example optimizations (eliminating useless and unreachable code, code motion, specialization,

enabling other transformation, redundancy elimination)., Advanced topics (Combining optimizations, strength reduction).

Unit 5: Instruction Selection & Scheduling [8 Hrs]
Introduction, Instruction selection and code generation via Sethi Ullman, Aho Johnson algorithm, Instruction selection via tree-pattern matching, Instruction selection via peephole optimization, Learning peephole patterns, Generating instruction sequences, Introduction to instruction scheduling, The instruction scheduling problem, List scheduling, Regional scheduling.

Unit 6:Register Allocation [6 Hrs]
Introduction, Issues in register allocation, Local register allocation and assignment, Moving beyond single block, Global register allocation and assignment, Variations on Graph Coloring Allocation, Harder problems in register allocation, CASE Study of GCC compiler.

Text Books

1. Keith D. Cooper and Linda Torczon, Engineering a Compiler, Elsevier-Morgan Kaufmann Publishers, 2004.
2. Steven S. Muchnick, Advanced Compiler Design Implementation, Elsevier-Morgan Kaufmann Publishers, 2003.
3. Uday Khedker, Amitabha Sanyal, Bageshri Karkare , Data Flow Analysis: Theory and Practice, CRC Press, 2009

Reference Books

1. Andrew Appel, Modern Compiler Implementation in C: Basic Techniques, Cambridge University Press, 1997.
2. Y.N. Srikant, Priti Shankar, The Compiler Design Handbook: Optimizations and Machine Code Generation, CRC Press, 2nd Edition, 2002.
3. David R. Hanson , Christopher W. Fraser, A Retargetable C Compiler: Design and Implementation, Addison-Wesley, 1995
4. Morgan, Robert, Building an Optimizing Compiler, Digital Press Newton, 1998.
5. John Levine, Tony Mason & Doug Brown, Lex and Yacc, O'Reilly

[COC(DE)-19011] Deep Learning

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:

1. Understand the fundamentals of neural networks.
2. Design feed forward networks with backpropagation.
3. Analyze neural networks for performance.
4. Apply attention mechanism to the neural network.

Unit 1: Basics [4 Hrs]

Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability. Convergence theorem for Perceptron Learning Algorithm.

Unit 2: Feedforward Networks **[6 Hrs]**
Introduction to neural network and multilayer perceptrons (MLPs) representation power of MLPs, sigmoid neurons, gradient descent, feed forward neural networks representation, Back-propagation.

Unit 3: Optimization Techniques **[8 Hrs]**
Gradient Descent, Batch Optimization, Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

Unit 4: Autoencoders **[10 Hrs]**
Autoencoders, Regularization in autoencoders, Denoising autoencoders, Sparse autoencoders, Contractive autoencoders, Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Greedy Layerwise Pre-training, Better activation functions, Better weight initialization methods, Batch Normalization.

Unit 5: Convolutional Neural Networks (CNN) **[8 Hrs]**
Introduction to CNN, Building blocks of CNN, Transfer Learning, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet, Visualizing CNNs, Guided Backpropagation, Fooling Convolutional Neural Networks.

Unit 6: Recurrent Neural Networks (RCNN) **[8 Hrs]**
Introduction to RCNN, Backpropagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs, Encoder Decoder Models, Attention Mechanism.

Text Book

1. Deep Learning- Ian Goodfelllow, Yoshua Benjio, Aaron Courville, The MIT Press

Reference Books

1. Neural Networks: A Systematic Introduction, Raúl Rojas, 1996
2. Pattern Recognition and Machine Learning, Christopher Bishop, 2007

[COC(DE)-19012] Multicore Technology

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Understand the working principles of multicore architectures.
2. Optimize performance of multicore systems.
3. Specify the necessity of GPU.
4. Comprehend and differentiate between CPU and GPU.
5. Identify and demonstrate the need of domain specific architectures.

Unit I :Introduction to Multicore Systems: [5 Hrs]

Fundamentals, The Era of Multicore Machines, Unicore vs Multicore - Understanding Performance - Shared Memory Multicore Systems - Distributed Memory Multicore Systems - Hybrid Systems - Symmetric and Asymmetric Multicore Systems – Overview of Multithreading – Multithreading in different forms – Homogeneous and Heterogeneous Multicore systems – Examples of different Multicore Systems.

Unit II :Cache Memory: [7 Hrs]

Large Cache Design: Shared vs. Private Caches - Centralized vs. Distributed - Shared Caches – Coherence: Snooping-based cache coherence protocol, directory-based cache coherence protocol - Uniform Cache Access, Non-Uniform Cache Access - S-NUCA, D-NUCA - Inclusion, Exclusion – Examples of different Cache Organization - Consistency Models – Case Study.

Unit III :Performance and Optimizations for Multicore Systems: [4 Hrs]

Select the right “core” - Improve serial performance - Achieve proper load balancing - Improve data locality - Reduce or eliminate false sharing - Use of affinity scheduling - Lock granularity and frequency - Remove synchronization barriers - Minimize communication latencies - Use of thread pools - Managing thread count - Use of parallel libraries.

Unit IV :Programming Multicore Systems: [8 Hrs]

Programming models for Multicore Systems – Shared Memory Programming using pthreads - Shared Memory Programming using OpenMP – Use of OpenMP compiler directives – #pragma with different clauses – Understanding parallelized loops – Synchronization Constructs towards dependencies – Function parallel program - OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, Performance.

Unit V :Special Case – Graphics Processing Unit: [7 Hrs]

CPU architecture - GPU hardware – CPU and GPU: Design Goals – Compute levels – Case Study: Nvidia GPU – GPGPU - Compute Unified Device Architecture (CUDA) Programming model – Applications of CUDA - Threads, Blocks, Grids – Memory management – Examples – Alternatives to CUDA.

Unit VI :Domain Specific-Architectures: [5 Hrs]

Guidelines for domain specific architectures – Deep Learning Architecture - Google’s Tensor Processing Unit (TPU) for Deep Neural Networks (DNNs) - Pixel Visual Core, a Personal Mobile Device Image Processing Unit.

Text Books and Research Papers

1. Gerassimos Barlas, “Multicore and GPU Programming: An Integrated Approach”, Morgan Kaufmann, 2015.
2. Rob Oshana, “Multicore Application Development Techniques: Applications, Tips and Tricks”, Elsevier, 2016.
3. John L Hennessy, David A Patterson, “Computer Architecture: A Quantitative Approach”, Sixth Edition, Morgan Kaufmann, 2018.

Reference Books

1. Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar, “Multi-Core Cache Hierarchies”, Morgan & Claypool Publishers, 2011.
2. Daniel J. Sorin, Mark D. Hill, David A. Wood “A Primer on Memory Consistency and Cache Coherence”, Morgan & Claypool Publishers, 2011.
3. Shane Cook, “CUDA Programming: A Developer’s Guide to Parallel Computing with GPUs”, Morgan Kaufmann, 2013.

[COC-19009] Data Mining and Machine Learning

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each
	End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Understand Supervised, unsupervised and semi supervised machine learning algorithm
2. Study of probabilistic analysis, parametric and non-parametric algorithms
3. Estimation of Maximum Likelihood, losses and risks for sample implementation
4. Study and Compare various classification, association, clustering algorithms
5. Apply data mining algorithms for solving real life problems
6. Discuss active areas of research in Data Mining and Machine Learning

Unit 1: Introduction [6 Hrs]

Introduction to data mining, Applications, Motivation, Data mining knowledge discovery process, kinds of data, data mining techniques, issues in data mining Introduction to Machine Learning: What is machine learning, Applications of ML, Design Perspective and Issues in ML, Supervised, Unsupervised, Semi-supervised learning with applications and issues.

Unit 2: Input, Output and Data Pre-processing [6 Hrs]

Input : Concepts, instances and attributes, Output: Knowledge Representation: Decision tables, Decision trees, Decision rules, Rules involving relations, Instance-based representation. Data Pre-processing-data cleaning, data integration and transformation, data reduction, data discretization and concept hierarchy generation.

Unit 3: Classification, Diagnostic and Prediction [8 Hrs]

Introduction to Classification, issues regarding classification, Classification: Model (or hypothesis) representation, decision boundary, cost function, gradient descent, regularization. Diagnostic: debugging a learning algorithm, evaluating a hypothesis

(Model selection), training/validating/testing procedures (offline and online training of models), diagnosing bias versus variance and vice versa, and bias/variance, learning curves, regularization techniques, Accuracy and Error measures: classifier accuracy measures, predictor error measure, evaluating the accuracy of a classifier or predictor, Confusion metric, precision, recall, tradeoff between both, accuracy, Analysis of ROC, AUC.

Unit 4: Decision tree, Probabilistic classifier, Clustering [6 Hrs]
Decision Tree: representation, hypothesis, issues in Decision Tree Learning, Pruning, Rule extraction from Tree, Learning rules from Data. Probabilistic classifier: Bayes rule, Maximum Likelihood Estimation, case study, Clustering :Unsupervised learning technique, Similarity and Distance Measures, k-means and k-medoids algorithm, optimization objective, random initialization, choosing value of k, EM algorithm.

Unit 5: Association Rule Mining and Support Vector Machines [6 Hrs]
Mining Frequent Patterns, Associations and Correlations: Basic concepts, Apriori algorithm for finding frequent itemsets using candidate generation, generating association rules from frequent itemsets, from association to correlation analysis Support Vector Machines: Objective(optimization), hypothesis, SVM decision boundary, kernels : RBF and others.

Unit 6: Advanced Techniques [6 Hrs]
Neural Networks, use case involving the use of neural network, role of various activation functions, SVD, Latent Dirichlet Allocation model, Latent Semantic Indexing, Models for Time-series forecasting - AR, MA, ARMA, ARIMA.

Text Books

1. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.
2. Jiawei Han Micheline Kamber, Data Mining Concepts and Techniques, Latest Edition

Reference Books

1. Ethem Alpaydin, Introduction to Machine Learning, PHI, 2005
2. D. Hand, H. Mannila and P. Smyth. Principles of Data Mining. Prentice-Hall. 2001
3. K.P. Soman, R. Longonathan and V. Vijay, Machine Learning with SVM and Other Kernel Methods, PHI-2009
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006.
5. M. H. Dunham. Data Mining: Introductory and Advanced Topics. Pearson Education. 2001
6. Witten, E. Frank, Mark Hall, C. Pal. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann. 2016
7. T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006. Link: https://people.inf.elte.hu/kiss/13dwhdm/roc.pdf

[COC-19010] Security in Computing

Teaching Scheme	**Examination Scheme**
Lectures: 3 hrs/week	T1, T2 – 20 marks each

End-Sem Exam – 60 marks

Course Outcomes

Students will be able to:

1. Demonstrate the importance of security in networked computing environments
2. Determine appropriate mechanisms such as encrypt, decrypt and transmit messages using cryptographic techniques for protecting networked systems.
3. Analyze high level the vulnerabilities and threats in the Internet and networked computing systems.
4. Develop basic system security mechanisms, such as those used in operating systems, file systems and computer networks
5. Discuss the issues concerning various threats to wireless networks, encryption and decryption

Unit 1: Introduction **[6 Hrs]**

Cryptography and Modern Cryptography, Basic concepts: threats, vulnerabilities, controls; risk; Security services, security policies, security mechanisms. Active vs. Passive attacks, Historical Ciphers and Their cryptanalysis, one time passwords (Vernam's Cipher)

Unit 2: Number Theory **[6 Hrs]**

Review of number theory and algebra, computational complexity, probability and information theory, primality testing, the Euclidean algorithm – Congruences: Definitions and properties – linear congruences, residue classes, Euler's phi function – Fermat's Little Theorem – Chinese Remainder Theorem

Unit 3: Symmetric Key Encryption **[6 Hrs]**

Cryptography and cryptanalysis, DES, Triple DES, AES, IDEA, CAST-128, RC4, Modes of operation.

Unit 4: Public Key Cryptography **[6 Hrs]**

RSA cryptosystem, Diffie-Hellman, Elliptic curve cryptography, Rabin, ElGamal, Goldwasser-Micali, Blum-Goldwasser cryptosystems.

Unit 5: Threats To Wireless Network **[6 Hrs]**

Wireless availability, Privacy Challenges. Risks: denial of Service, Insertion Attacks, interception and monitoring wireless traffic, Mis-configuration. Wireless Attack: Surveillance, War Driving, Client-to-Client Hacking, Rogue Access Points, Jamming and Denial of Service.

Unit 6: Wireless Network Security **[6 Hrs]**

Access Point-Based Security Measures, Thin Party Security Methods, Funk's Steel-Belted Radius, VVLAN Protection Enhancements, Blue-tooth Security Implementation, Security in WIMAX, UWB security, Satellite network security

Text Books

1. William Stallings, Cryptography and Network Security, Prentice Hall, 4th Edition, 2006
2. Behrouz A Forouzan, Cryptography & Network Security, McGraw-Hill, 2008

3. Atul Kahate, Cryptography and Network Security, Tata McGraw-Hill, 2nd Edition, 2008.
4. William Stallings, Network Security Essentials Applications and Standards, Pearson Education, New Delhi.

Reference Books

1. C. Pfleeger and S. Pfleeger, Security in Computing, Prentice Hall,4th Edition, 2007.
2. Eric Maiwald, Fundamentals of Network Security, McGraw-Hill, 2004.
3. Jay Ramachandran, Designing Security Architecture Solutions, Wiley Computer Publishing, 2002.
4. Bruce Schneier, Applied Cryptography, John Wiley & Sons Inc, 2001.
5. Charlie Kaufman, Radia Perlman and Mike Speciner, Network Security Private Communication in a public world, Prentice Hall of India Private Ltd., New Delhi

[COC-19011] Embedded Systems

Teaching Scheme
Lectures: 3 hrs/week

Examination Scheme
T1, T2 – 20 marks each
End-Sem Exam – 60 marks

Course Outcomes
Students will be able to:
1. Explain Characteristics & Salient Features of Embedded Systems
2. Analyze Architecture & Recent Trends of Embedded Systems
3. Discuss PIC and ARM families
4. Understand general process of embedded system development and implement them.
5. Explain communication interface for wired and wireless protocols
6. Discuss hardware and software design methodologies for embedded systems

Unit 1: Overview of Embedded Systems [4 Hrs]
Introduction, Definition, Characteristics & Salient Features, Classification, Application Areas, Overview of Embedded System Architecture & Recent Trends.

Unit 2: Hardware Architecture [8 Hrs]
Embedded Hardware based on Microprocessors, Microcontrollers & DSPs. Study of PIC Microcontrollers: PIC16C6X/7X Family & Applications. Study of ARM Family : ARM 7,9,10 &11: Overview & Architecture Comparison, Detailed Study of ARM7-TDMI including Core Architecture, ARM/Thumb State, On Chip Debug & Development Support, AMBA Bus, Applications.

Unit 3: Communication Interface [6 Hrs]
Serial, Parallel, Wired Wireless Protocols Wired : CAN ,I2C,USB, FireWire Wireless : Blue Tooth , IrDA, IEEE802.11.

Unit 4: Software Architecture [6 Hrs]

Concepts: Embedded OS, Real-Time Operating Systems (RTOS), Detailed Study of RT Linux ,Hand Held OS, Windows CE. & Development Tools.

Unit 5: Embedded Systems for Automotive Sector [6 Hrs]
Electronic Control Units (ECU) for Engine Management, Antilock Braking System (ABS), Crusie Control, Design Challenges, Legislative Emission Norm, Interface Standards, Developmental Tools Navigation Systems : Global Positioning System (GPS):Detailed Study & Applications.

Unit 6: [4 Hrs]
Smart Cards: Classifications, Interfacing, Standards & Applications.
RFID Systems: Technology, RFID Tag ,RFID Reader, Applications.

Unit 7: Case Studies [6 Hrs]
Embedded System for Mobile Applications, DSP Based Embedded System, Networked Embedded System & Digital Camera.

Text Books

1. K.V.K. Prasad, Embedded / Real Time Systems: Concepts, Design and Programming Black Book, Dreamtech Press, 2005.

Reference Books

1. Vahid F. and Givargies T., Embedded Systems Design, John Wiley X. Sons, 2002
2. John B Peatman, Design with PIC Microcontrollers, Pearson Education, 1998
3. Liu, Real-Time Systems, Pearson Education, 2000.
4. Technical Manuals of ARM Processor Family available at ARM Website on Net

[COC-19012] Data Mining and Machine Learning Laboratory

Teaching Scheme	**Examination Scheme**
Practical: 2 hrs/week	Term Work: 50 marks
	Oral Examination: 50 marks

Course Outcomes
Students will be able to:

1. Apply data preprocessing techniques on any data.
2. Apply all possible evaluation measures and diagnoses required on kinds of data
3. Apply learning techniques like classification, decision tress, naive bayesian model, clustering, SVM, ANN, etc., to solve a real-life problem.
4. Demonstrate the ability to analyze different machine learning algorithms using exploratory data analysis techniques
5. Compare and contrast the strengths and weaknesses of various Machine Learning approaches

Suggested list of Assignments

1. Take any benchmark dataset (both numeric and text) and apply preprocessing techniques on it.

2. Compare the performance of 10 machine learning models for regression data set (eg. UCI repository Breast Cancer dataset) for the data partition of 70-30% with acceptable error of ±100. The comparative study of machine learning models should be of the form:

Model	Method	Package	r	R2	Error	Accuracy
M1						
M2						
…						
M10						

3. Study 5 feature selection techniques on the regression data set considered in (2) and report top five features. The study of feature selection techniques should be represented as :

Feature Selection Technique	Tops 5 Features
T1	
T2	
..	
T5	

4. Estimate the accuracy of the Naive Bayes classifier on the breast cancer data set using 5-fold cross-validation.
5. Implement the SVM algorithm with RBF. Estimate the precision, recall, accuracy, and F-measure on the text classification task for each of the 10 categories using 10-fold cross-validation of Reuters dataset.
6. Implement both the *k*-means algorithm and the Hierarchical Agglomerative Clustering (HAC) algorithm. For both, assume that all features are real-valued. Also assume that there is no need for normalization of the features. Use the L_n-norm for distance calculations with a default value of n=2 (Euclidean).
   Note that the datasets you are to test your algorithms with contain labeled items. You will need to ignore the label (target attribute, always last here) while clustering.
7. Implement the k-means clustering algorithm and the HAC algorithm (using single linkage). For k-means, your program should automatically try 2 <= k <= 7 and compute the squared error in each case. You should then return the value of k that produces the lowest squared error, together with that error. For HAC, you should compute (and store) the squared errors of all possible clustering's (as they are built from the bottom up). Upon completion, you should return the value of t (distance threshold) that produces the lowest squared error, the corresponding number of clusters, and the corresponding error.
   Use the algorithms on the Iris dataset(available on UCI repository) .
   Compare the best number of clusters obtained by *k*-means and HAC. How do these also compare with the underlying structure of the dataset in which there are 3 classes of iris plants?
   Experiment with your distance metric -- can you find a value of *n* for the L_n-norm that changes the number of clusters found?
   Graph the value of the squared error for each clustering as HAC executes (with no threshold). What do you observe? Is this surprising?
8. Implement the Apriori algorithm. Build your own association task. Design your task so that it contains some simple associations you can check your algorithm

against. List these associations. Run Apriori for various combinations of minsup and minconf values. Verify that the associations you designed into the task are discovered by your algorithm.

Note: The teacher taking the course can change the list of assignments. This is just a guideline of list of assignments.

[COC-19013] Security in Computing Laboratory

Teaching Scheme:
Practical: 2 hrs/week

Examination Scheme:
Term Work: 50 marks
Oral Examination: 50 marks

Course Outcomes

Students will be able to:

1. Design own encryption technique using the concepts they studied
2. Demonstrate the practical importance of security in computing
3. Analyze the implementations for time required to generate keys and encryption/decryption process also various possible attacks
4. Install and configuring the proxy server, Firewall and IDS

Suggested list of assignments

1. Design and Implement your own encryption/ decryption algorithm using any programming language
2. Design an experiment to estimate the amount of time to Generate key pair (RSA): Encrypt n bit message (RSA), Decrypt n bit message (RSA), As function of key size, experiment with different n-bit messages. Summarize your conclusion.
3. Implementation of email security using PGP (create yourself a 1024-bit PGP key. Use your name and email address for your key label. Use PGP to verify the signature on this assignment.
4. Install any Proxy Server and configure an application gateway.
5. Install any Firewall and configure it as per the defined security policy.
6. Install, Configure and study any Intrusion Detection System (IDS).

[COC-19014] Embedded Systems Laboratory

Teaching Scheme:
Laboratory: 2 Hrs/week

Examination Scheme:
Continuous Evaluation : 50 marks
End-Sem Exam: 50 marks

Course Outcomes

Students will be able to:

1. Design a system using the concepts of RISC architecture and ARM processors
2. Decide the hardware and/or software components best suited for a given problem taking into consideration constraints such as cost, performance etc.
3. Develop prototype codes using commonly available on and off chip peripherals with and without interrupts on Cortex M3/M4 development boards
4. Develop a system using the concepts of RISC architecture and ARM processors

Suggested list of Assignments

Experiments to be carried out on Tiva (TM4C123X) Launch-pads:

1. Blink an LED with software delay, delay generated using the SysTick timer.
2. System clock real time alteration using the PLL modules.
3. Control intensity of an LED using PWM implemented in software and hardware.
4. Control an LED using switch by polling method, by interrupt method and flash the LED once every five switch presses.
5. Key matrix and alphanumeric LCD interfacing and programming.
6. UART programming with accessing TX ad RX buffers directly and using DMA.
7. Recording of analog readings at the output of rotary potentiometer connected to ADC channel.
8. Programming (ISL 29023) Ambient and Infrared Light sensor available on Sensor Hub Booster Pack using I^2C interface.
9. Calling C functions from assembly programs and vice versa.

[COC-19015] Dissertation Phase – I

Teaching Scheme:	**Examination Scheme:**
--- NA ---	End-Sem Evaluation: 100 marks

Course Outcomes

Students will be able to:

1. Demonstrate how the available literature can be searched for gathering information about a problem/domain
2. Identify the current status of the technology/research in the selected domain and open problems in the selected domain that have relevance to societal / industrial needs
3. Apply software engineering principles related to requirements gathering and analysis
4. Evaluate different design techniques and methods to find out the best feasible solution under given constraints for the chosen problem for dissertation
5. Deliver/produce artifacts such as requirements specification, detailed design etc.

Guidelines

The dissertation is a yearlong activity, to be carried out and evaluated in two phases. The dissertation may be carried out in-house or in industry allotted through the department. The project topic and internal guide (faculty member of the department) are decided at the beginning of Phase-I.

Student is expected to complete the following activities in Phase-I:

1. Literature survey
2. Problem Definition
3. Motivation for study and Objectives
4. Preliminary design / feasibility / modular approaches

Deliverables

1. A report having following details: Abstract, Problem statement, Requirements specification, Literature survey, Proposed solution, High level design description, Plan for implementation and testing in Phase-II
2. A presentation that covers the major points covered in the report

3. A proof of concept (preferable but not mandatory)

Evaluation

Two independent assessments will be done:

1. Internal guide will evaluate his/her student for 40 marks
2. A panel of External Examiner(s) and two senior faculty of the department will evaluate the work for 60 marks

The marks obtained in these two assessments will be combined to get final evaluation out of 100 marks. The grading, like other courses, will be relative.

The evaluation will take place based on criteria such as literature survey and well defined project problem statement, proposed high level system design, concrete plan for implementation and result generation, presentation etc.

The panel (external examiner(s) and senior faculty) will provide a report about suggestions/changes to be incorporated during phase-II.

[COC-19016] Massive Open Online Course – I

To be selected in consultation with the faculty advisor. Evaluation scheme will depend upon the instructor or host institute.

[COC-19017] Massive Open Online Course – II

To be selected in consultation with the faculty advisor. Evaluation scheme will depend upon the instructor or host institute.

[COC-19018] Dissertation Phase – II

Teaching Scheme:
--- NA ---

Examination Scheme:
Midterm Evaluation : 50 marks
End-Sem Evaluation: 50 marks

Course Outcomes

Students will be able to:

1. Apply project planning principles and techniques for effective and efficient project execution
2. Apply software engineering principles related to implementation and testing of software solutions
3. Demonstrate familiarity with the entire lifecycle of a software product/solution
4. Get proficiency in the language(s)/tool(s)/libraries/technology used in the dissertation work
5. Deliver/produce artifacts such as code, test plan, test results, research paper(s) based on the dissertation work etc.
6. Demonstrate presentation skills required to present the work done in various forms (technical report/paper/presentation) at various platforms (conferences/journals/defense of the dissertation etc)

Guidelines
Student is expected to complete the following activities in Phase-II:

1. Implementation of the proposed approach in the first stage
2. Testing and verification of the implemented solution
3. Writing of a report and presentation
4. Publish the work done at suitable conference/in a journal

Deliverables

1. Code (if the project is in-house)
2. Dissertation report that gives overview of the problem statement, literature survey, design, implementation details, testing strategy and results of testing
3. All the artifacts created throughout the duration of dissertation such as requirements specification, design, project plan, test cases etc
4. Presentation based on the dissertation report
5. Research Paper(s) based on the dissertation work

Evaluation

Evaluation will be done in two steps: mid-term evaluation and final evaluation.

1. Mid-term evaluation
   Evaluation will be done by the internal guide and a qualified external examiner
   The internal guide will evaluate his/her student for 20 marks.
   External Examiner will provide evaluation for 30 marks

   The assessment is done on the criteria such as concrete system design, implementation status and concrete plan for completion of remaining tasks, presentation etc.

   The purpose of mid-term evaluation is to check preparedness of students for the final evaluation. External examiner may give suggestions for changes/corrections to be incorporated before the final evaluation. If the work done till then may not lead to successful completion of the dissertation in the remaining time, student can be asked to take extension.

2. Final Evaluation
   Internal guide and one external examiner will carry out the final evaluation. The guide will provide evaluation for 20 marks and the external examiner for 30 marks.

   The assessment will be done based on the criteria such as quality of implementation,
   result analysis, project outcomes (publications, patent, copyright, contribution to open source community, participation in project competition etc.), quality of report, presentation etc.

The total assessment of phase-II work is for 100 marks (mid term evaluation for 50 marks and final evaluation for 50 marks) and the grading, like other courses, will be relative.