

Course Directory for

M. Tech. in Computer Science and Engineering

(Students admitted from 2015–16)

Offered by the

SCHOOL OF COMPUTING

This scheme was approved in the XXVIII Academic Council Meeting held on 06.06.2015

**Scheme of Study**

**I Semester (25 Credits)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Name** | **No of Contact Hours / Week** | | | **Total** |
| **L** | **T** | **P** |
| MCSE101R04 | Logic for Computer Science Engineers | 4 | 0 | 0 | 4 |
| MCSE102R04 | Advanced Data Structures & Algorithms | 4 | 0 | 0 | 4 |
| MCSE103R04 | High Performance Scientific Computing | 3 | 1 | 0 | 4 |
| MCSE104R01 | Network Programming & Security | 3 | 0 | 0 | 3 |
| MCSE105 | Virtualization Technologies | 4 | 0 | 0 | 4 |
| MCSE106R01 | Advanced Database Design | 4 | 0 | 0 | 4 |
| MCSE107R02 | Programming Lab - I | 0 | 0 | 4 | 2 |
| **TOTAL** | | **22** | **1** | **4** | **25** |

**II Semester (26 Credits)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Name** | **No of Contact Hours / Week** | | | **Total** |
| **L** | **T** | **P** |
| MCSE201RXX | Statistical Data Analysis | 4 | 0 | 0 | 4 |
| MCSE202RXX | Cloud Computing | 4 | 0 | 0 | 4 |
| MCSE203RXX | Software Testing & Automation | 3 | 1 | 0 | 4 |
| MCSE204RXX | Web Services | 3 | 0 | 0 | 3 |
| MCSE205EXX | Elective – I | 4 | 0 | 0 | 4 |
| MCSE206EXX | Elective – II | 4 | 0 | 0 | 4 |
| MCSE207RXX | Programming Lab - II | 0 | 0 | 4 | 2 |
| MCSE208 | Seminar | 0 | 0 | 2 | 1 |
| **TOTAL** | | **22** | **1** | **6** | **26** |

**III Semester (18 Credits)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Name** | **No of Contact Hours / Week** | | | **Total** |
| **L** | **T** | **P** |
| MCSE301RXX | Multi Agent Systems | 4 | 0 | 0 | 4 |
| MCSE302EXX | Elective – III | 4 | 0 | 0 | 4 |
| MCSE303EXX | Elective – IV | 4 | 0 | 0 | 4 |
| MCSE304 | Project - Phase I | 0 | 0 | 12 | 6 |
| **TOTAL** | | **12** | **0** | **12** | **18** |

**IV Semester (12 Credits)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course Name** | **No of Contact Hours / Week** | | | **Total** |
| **L** | **T** | **P** |
| MCSE401 | Project - Phase II | 0 | 0 | 24 | 12 |
| **TOTAL** | | **0** | **0** | **24** | **12** |

**Total credits: 25 +26 +18 +12 = 81 credits**

**List of Electives**

|  |
| --- |
| 1. Modern Embedded Computing |
| 1. Computer Vision & Machine Learning |
| 1. Wireless Sensor Networks |
| 1. Fault Tolerant Computing |
| 1. Image Processing & Pattern Recognition |
| 1. Mobile & Ubiquitous Computing |
| 1. Software Architecture & Design Patterns |
| 1. Semantic Web & Social Networks |
| 1. Big Data Analytics |
| 1. Internet of Things |
| 1. Game Theory |
| 1. Advanced Computer Architecture |
| 1. Randomized Algorithms |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE101R04 Semester: I**

**LOGIC FOR COMPUTER SCIENCE ENGINEERS**

**Course Objectives::**

To help learners understand the basic principles of a wide variety of logic and put its use in practical real world applications.

**UNIT - I 12 Periods**

**Propositional and First Order Logic:** Propositional Logic - Principles of Satisfiability, Validity, Soundness and Completeness - Consistency - Strong Completeness and Compactness - Davis Putnam Algorithm - DPLL algorithm - Usage of DPLL algorithm for solving real world problems - Improving DPLL algorithm - Stochastic algorithms - Complexity of SAT - First Order Logic - Gentzen System - Hilbert System - Equivalence - C-Rule - Skolem theorem.

**UNIT - II 12 Periods**

**First Order Logic and Pi Calculus:** First Order Logic Normal Forms - Herbrand Models - Theorem - Ground Resolution - Substitution - Unification - Soundness and Completeness of General Resolution - Horn Clauses - Undecidability and Model Theory of First Order Logic.

Pi Calculus - Inductive Principles - Asynchronous Pi Calculus - Reduction semantics - Action semantics - Justifying bisimulation equivalence.

**UNIT - III 12 Periods**

**Verification of sequential and Concurrent programs using Temporal Logic:** Temporal Logic - Introduction - Properties - Linear Temporal Logic - Deductive System - Theorems of L - Soundness and Completeness of L - Verification of Sequential Programs - verification of Concurrent programs - Formalization of Correctness - Deductive Verification of Concurrent Programs - Programs as Automata - Model Checking of Invariance Properties - Model Checking of Liveness Properties - Expressing an LTL Formula as an Automaton - Model Checking Using the Synchronous Automaton - Branching-Time Temporal Logic - Symbolic Model Checking.

**UNIT - IV 12 Periods**

**Lambda Calculus:** Lambda Calculus - Combinatory Logic - Power of Lambda and Combinators - Representing computable functions - Undecidability theorem.

**UNIT - V 12 Periods**

**Application of Lambda calculus in Functional Programming** : Functional Programming in Standard ML - Types - Lists - Tuples - Function types and expressions - Standard Functions - Comparison Operators - Recursion - Tuple selection - Pattern matching - Type expressions - Variables and Polymorphism - New Types - Trees - Lambda calculus in SML - Functional Programming and LISP - Applications of pi calculus in modeling a firewall

**REFERENCES**

1. Mordechai Ben-Ari. *Mathematical Logic for Computer Science*. Springer Publications, Third Edition, 2012.
2. Greg Michaelson. *An Introduction to Functional Programming through Lambda Calculus.* Dover Publications, 2011.
3. J. Roger Hindley, Jonathan P. Seldin. *Lambda-Calculus and Combinators, An Introduction.* Cambridge University Press, Second Edition, 2008.
4. Mathew Hennessy. *A Distributed Pi Calculus.* Cambridge University Press, 2007.
5. Henk Barendregt, Erik Barendsen. *Introduction to Lambda Calculus.* Revised Edition, March 2000.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to contemplate various properties associated with logic and be able to solve associated problems |
| Unit II | The learner will be able to model using first order logic and pi calculus |
| Unit III | The learner will be able to analyze sequential and concurrent programs using temporal logic |
| Unit IV | The learner will be able to solve problems using lambda calculus |
| Unit V | The learner will be able to use Lambda calculus and Pi calculus in solving real world problems |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE102R04 Semester: I**

**ADVANCED DATA STRUCTURES & ALGORITHMS**

**Course Objectives:**

To understand efficient data representation methods and various algorithmic strategies for effective utilization of memory and processing time.

**UNIT - I 12 Periods**

**Introduction**: Role of algorithms in computing - Growth of functions: Asymptotic notations -Standard Notations and common functions - Summation formulas and Properties - Bounding Summations - Methods for solving recurrences: Substitution, Recursion tree, Master methods.

**Hash tables**: Direct Address tables - Hash Tables - Hash Functions - Open Addressing - Perfect Hashing - Skip Lists - Splay Trees - AVL trees - Red-Black trees - 2-3 trees.

**UNIT - II 12 Periods**

**B-Trees**: Basic operations on B-Trees - Deletion a key from a B-Tree - Fibonacci Heaps - Structure of Fibonacci Heap - Mergeable-heap operation - Decreasing a key and Deleting a node - Bounding the maximum degree - van Emde Boas Trees - approaches - recursive structures.

**Data Structures for Disjoint Sets**: Disjoint-set operations - Linked list representation - Disjoint set forests.

**Data Structures for strings**: Tries and Compressed Tries - Dictionaries allowing errors in Queries - suffix trees - Suffix Arrays.

**UNIT - III 12 Periods**

**Graph Algorithms:** Representation of Graphs - BFS - DFS - Correctness and Complexity Analysis - Topological sort - Strongly Connected Components - Minimum Spanning Trees -Correctness and Complexity Analysis of Kruskal and Prim algorithms - Single Source Shortest Paths - Bellman-Ford algorithm - Single Source shortest paths in directed acyclic graphs - Dijkstra’s Algorithm - Difference constraints and shortest paths - proofs of shortest path properties - All-Pairs Shortest Paths - Shortest paths and matrix multiplication - Floyd-Warshall algorithm - Johnson’s algorithm for sparse graphs.

**UNIT - IV 12 Periods**

**Maximum Flow**: Flow Networks - Ford Fulkerson method - Minimum Bipartite matching - Push-relabel algorithms.

**String Matching Algorithms**: Naïve Matching - Rabin-Karp - String matching with finite automata - Knuth-Morris-Pratt algorithm –

**Computational Geometry**: Line-segment properties - Determining whether any pair of segments intersects - Finding the convex hull - Finding the closest pair of points.

**UNIT - V 12 Periods**

**NP-Completeness**: Polynomial time - Polynomial-time veriﬁcation - NP-completeness and reducibility - NP-completeness Problems.

**Approximation Algorithms**: Vertex Cover Problem - Traveling Salesman Problem - Set Covering Problem - The subset-sum problem.

**REFERENCES**

1. Thomas H. Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein. *Introduction to Algorithms*. MIT Press, Third Edition, 2014.
2. Peter Brass. *Advanced Data Structures.* Cambridge University Press, First Edition, 2008.
3. Jon Kleinberg, Eva Tardos. *Algorithm Design.* Addison Wesley, Pearson Education, First Edition, 2005.
4. Mark Allen Weiss. *Data Structures and Algorithm Analysis in C++.* Addison Wesley, Pearson Education, Fourth Edition, 2014.
5. Thomas H. Cormen. *Algorithms Unlocked*. MIT Press, 2013.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the representation of algorithm complexity and the representation and manipulation of hash table, binary search tree, and red black tree |
| Unit II | The learner will be able to explore the different kinds of trees, their manipulations and applications |
| Unit III | The learner will be able to understand the representation of graph and their applications in various areas |
| Unit IV | The learner will be able to apply the graph algorithm in flow networks, design efficient pattern matching algorithms for strings, and comprehend various algorithms used in computational geometry |
| Unit V | The learner will be able to investigate the various NP-Complete problems and approximation algorithms for solving them |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **3** | **1** | **0** | **4** |

**Course Code: MCSE103R04 Semester: I**

**HIGH PERFORMANCE SCIENTIFIC COMPUTING**

**Course Objectives::**

To understand the demand for computational speed, various types of parallel computers and cluster computing, apply various parallel programming techniques to solve simple problems like sorting, prime number generation, LU factorization, graph partitioning etc., understand the basic concepts of synchronization and dynamic load balancing

**UNIT - I 10 Periods**

**Parallel computers** - Basic techniques - Demand for computational speed - Potential for increased computational speed - Types of parallel computers - Cluster computing.

**UNIT - II                                                                                       12 Periods**

**Parallel computations** - Pipeline techniques - Memory hierarchy and pipelines - Computing platform for pipelined applications - Pipeline program examples: adding numbers, sorting numbers, prime number generation.

**UNIT - III                                                                                                        12 Periods**

**Basics of programming** using Message Passing Interface - Debugging and evaluating parallel programs - Basic linear algebra: Using BLAS - Vector norms and matrix norms-Parallel matrix multiplication: Triple nested for-loop algorithm, block matrix multiplication algorithms, BMR algorithm - Parallel LU factorization - Parallel triangular solve - Sparse matrices - Parallel direct solvers for sparse matrices.

**UNIT - IV                                                                                    12 Periods**

**Partitioning** - Partitioning strategies - Divide and conquer examples: sorting using bucket sort -Heuristic graph partitioning using Kernighan-Lin algorithm - Multilevel graph partitioning using Metris - Numerical integration - Synchronous computations - Barriers: counter based approach, tree based approach, butterfly approach.

**UNIT - V                                                                                 14 Periods**

**Data parallel computations** - Synchronous iterations - Solving a system of linear equations by iteration-Conjugate Gradient method-Dynamic load balancing - Searching a graph - Shortest path problem.

**REFERENCES**

1. Barry Wilkinson and Michael Allen. *Parallel Programming Techniques and Applications using Networked Workstations and Parallel Computers.* Prentice Hall, Second Edition,2005
2. Kai Hwang, Zhiwei Xu. *Scalable Parallel Computing Technology, Architecture, Programming.* McGraw Hill International, 2000.
3. David. E. Culler, Jaswinder Pal Singh, Anoop Gupta. *Parallel computer Architecture, A Hardware or Software approach.* Morgan Kaufmann Publication, First Edition, 1999.
4. Michael J. Quinn. *Parallel Computing, Theory & Practice.* McGraw Hill, Second Edition, 1994.
5. Ian Foster. *Designing and Building Parallel Programs*. Addison-Wesley, 2004.

**LEARNING OUTCOME**S

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the basics of parallel computation, types of parallel computers, computational speed and cluster computers |
| Unit II | The learner will be able to understand basics of pipeline design and apply the same to generate prime numbers and to sort a list of numbers |
| Unit III | The learner will be able to understand the basics of MPI and BLAS. The learner will apply MPI to solve problems like Matrix multiplication and LU factorization. Analyse the performance of the Matrix operations in MPI and BLAS |
| Unit IV | The learner will be able to apply the divide and conquer technique to perform numerical integration and graph partition. They will also understand the basic concepts of barriers in parallel computation |
| Unit V | The learner will be able to apply the data parallelization techniques to search a given graph, solve a system of linear equations and computing the shortest path between two nodes in a graph |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **3** | **0** | **0** | **3** |

**Course Code: MCSE104R01 Semester: I**

**NETWORK PROGRAMMING & SECURITY**

**Course Objectives:**

To help the learner familiarize Network Security features and develop applications using socket programming in Unix environment.

**UNIT - I 9 Periods**

**Introduction and TCP/IP:** Introduction - A Simple Daytime Client - Protocol Independence - Error Handling: Wrapper Functions - A Simple Daytime Server - Roadmap to Client/Server Examples in the Text - OSI Model - BSD Networking History - Test Networks and Hosts - Unix Standards - 64-Bit Architectures.

**The Transport Layer:** TCP, UDP, and SCTP - Introduction - The Big Picture - User Datagram Protocol (UDP) - Transmission Control Protocol (TCP) - Stream Control Transmission Protocol (SCTP) - TCP Connection Establishment and Termination - TIME\_WAIT State - SCTP Association Establishment and Termination - Port Numbers - TCP Port Numbers and Concurrent Servers - Buffer Sizes and Limitations - Standard Internet Services - Protocol Usage by Common Internet Applications.

**UNIT - II 9 Periods**

**Elementary Sockets:** Sockets Introduction - Socket Address Structures - Value-Result Arguments - Byte Ordering Functions - Byte Manipulation Functions - inet\_aton, inet\_addr, and inet\_ntoa Functions - inet\_pton and inet\_ntop Functions - sock\_ntop and Related Functions - readn, writen, and readline Functions.

**Elementary TCP Sockets:** TCP Sockets Introduction - socket, connect, bind,listen, accept,fork and exec Functions - Concurrent Servers - close Function - getsockname and getpeername Functions.

**TCP Client/Server Example:** Introduction - TCP Echo Server: main and str\_echo Function - TCP Echo Client: main str\_cli Function - Normal Startup - Normal Termination - POSIX Signal Handling - Handling SIGCHLD Signals - wait and waitpid Functions - Connection Abort before accept Returns - Termination of Server Process - SIGPIPE Signal - Crashing of Server Host - Crashing and Rebooting of Server Host - Shutdown of Server Host - Data Format.

**UNIT - III 9 Periods**

**I/O Multiplexing:** The select and poll Functions Introduction - I/O Models - select Function - str\_cli Function - Batch Input and Buffering - shutdown Function - pselect and poll Function.

**Socket Options:** Introduction - getsockopt and setsockopt Functions - Checking if an Option Is Supported and Obtaining the Default - Socket States - Generic Socket Options - IPv4 Socket Options - ICMPv6 Socket Option - IPv6 Socket Options - TCP Socket Options - SCTP Socket Options - fcntl Function.

**Elementary UDP Sockets:** Introduction - recvfrom and sendto Functions - UDP Echo Server: mainand dg\_echo Function - UDP Echo Client: main and dg\_cli Function - Lost Datagrams - Verifying Received Response - Server Not Running - connect Function with UDP - Lack of Flow Control with UDP - Determining Outgoing Interface with UDP - TCP and UDP Echo Server Using select.

**UNIT - IV 9 Periods**

**Introduction:** Computer Security Concepts - The OSI Security Architecture - Security Attacks - Security Services - Security Mechanisms - A Model for Network Security - Standards.

**Symmetric Encryption and Message Confidentiality:** Symmetric Encryption Principles - Symmetric Block Encryption Algorithms - Random and Pseudorandom Numbers - Stream Ciphers and RC4 - Cipher Block Modes of Operation - Recommended Reading and Web Sites - Key Terms, Review Questions, and Problems.

**Public-Key Cryptography and Message Authentication:** Approaches to Message Authentication - Secure Hash Functions - Message Authentication Codes - Public-Key Cryptography Principles - Public-Key Cryptography Algorithms - Digital Signatures.

**UNIT - V 9 Periods**

**Key Distribution and User Authentication:** Symmetric Key Distribution Using Symmetric Encryption - Kerberos - Key Distribution Using Asymmetric Encryption - X.509 Certificates - Public-Key Infrastructure - Federated Identity Management.

**Transport-Level Security:** Web Security Considerations - Secure Socket Layer and Transport Layer Security - Transport Layer Security - HTTPS -Secure Shell (SSH).

**Wireless Network Security:** IEEE 802.11 Wireless LAN Overview - IEEE 802.11i Wireless LAN Security - Wireless Application Protocol Overview - Wireless Transport Layer Security - WAP End-to-End Security.

**REFERENCES**

1. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff. *UNIX Network Programming.* Volume 1, Addison Wesley, Third Edition, 2003.
2. William Stallings. *Network Security Essentials: Applications and Standards.* Pearson Education, Fourth Edition, 2011.
3. W. Richard Stevens. *UNIX Network Programming Volume 2: Inter process Communications.* Addison Wesley, Second Edition, 1998.
4. William Stallings. *Cryptography and Network Security: Principles and Practice.* Pearson Education, Fifth Edition, 2011.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the basic concepts of TCP/IP and Transport layer |
| Unit II | The learner will be able to create application using elementary TCP sockets |
| Unit III | The learner will be able to understand the basic concepts of I/O multiplexing and create application using elementary UDP sockets |
| Unit IV | The learner will be able to understand concepts of security services and algorithms |
| Unit V | The learner will be able to gain an understanding of the security principles, IP Security, systems along with authentication and authorization techniques |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE105 Semester: I**

**VIRTUALIZATION TECHNOLOGIES**

**Course Objectives:**

To help the learner to understand virtual machine technologies in a number of contexts to enable new capabilities and to solve a variety of problems in interfacing major computer system components

**UNIT - I 12 Periods**

**Overview of Virtualization:** Virtualization - Basic approaches to Virtual systems - Advantages of Virtualization - Virtualization Caveats - A Model of Virtualization - Access Virtualization - Application Virtualization - Processing Virtualization - Network Virtualization - Storage Virtualization - Security for Virtual Environments - Management for Virtual Environments.

**UNIT - II 12 Periods**

**Computer Architecture** - Process Virtual Machines - System Virtual Machines.

**Emulation: Interpretation and Binary Translation:** Basic Interpretation - Threaded Interpretation - Predecoding and Direct Threaded Interpretation - Interpreting a Complex Instruction Set - Binary Translation - Code Discovery and Dynamic Translation - Control Transfer Optimizations - Instruction Set Issues.

**UNIT - III 12 Periods**

**Process Virtual Machines:** Virtual Machine Implementation - Compatibility - State Mapping - Memory Architecture Emulation - Instruction Emulation - Exception Emulation - Operating System Emulation-Code Cache Management - System Environment.

**Dynamic Binary Optimization:** Dynamic Program Behavior - Profiling-Optimizing Translation Blocks -Optimization Framework - Code reordering-Code Optimizations.

**High-Level Language Virtual Machine Implementation:** Object-Oriented High - Level Language Virtual Machines - The Java Virtual Machine Architecture - Completing the Platform: APIs

**UNIT - IV 12 Periods**

**Code-signed Virtual Machines:** Memory and Register State Mapping-Self-Modifying and Self-Referencing Code - Support for Code Caching-Implementing Precise Traps - Input/Output Applying Code-signed Virtual Machines - Case Study: Transmeta Crusoe.

**System Virtual Machines:** Key Concepts - Resource Virtualisation - Processors-Resource Virtualisation - Memory-Resource Virtualisation-Input/Output - Performance Enhancement of System Virtual Machines-Case Study: VMware Virtual Platform.

**UNIT - V 12 Periods**

**Multiprocessor Virtualization:** Partitioning of Multiprocessor Systems - Physical Partitioning - Logical Partitioning - Case Study: Cellular Disco System Virtual Machine - Based Partitioning-Virtualization with Different Host and Guest ISAs.

**Emerging Applications:** Security - Migration of Computing Environments - Grids: Virtual Organisations.

**REFERENCES**

1. Dan Kusnetzky. *Virtualization: A Manager’s Guide*, O’Reilly Media, Inc., First Edition, 2011.
2. William Von Hagen. *Professional Xen Virtualization.* Wiley Publishing Inc., First Edition, 2009.
3. James E. Smith, Ravi Nair. *Virtual machines Versatile platforms for systems and processes*. Morgan Kaufmann Publishers, First Edition, 2005.
4. Danielle Ruest and Nelson Ruest. *Virtualization: A Beginner’s Guide*. McGraw-Hill, First Edition, 2009.
5. Ivanka Menken. *Virtualization: The Complete Corner Stone Guide to Virtualization Best Practices*. McGraw-Hill, 2009.
6. Sander van Vugt. *A Practical Guide to XEN High Availability Configuring Enterprise Virtualization on SUSE Linux Enterprise Server.* Books4Brains, 2010.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the concepts behind virtualization technology, the different categories of virtualization, how they are used and merits & demerits Virtualization |
| Unit II | The learner will be able to classify different types of virtual machines and issues related to the emulation of a source instruction set architecture (ISA) on a target ISA |
| Unit III | The learner will be able to comprehend the implementation of process virtual machines and examines the techniques for the optimization of translated code for better emulation performance |
| Unit IV | The learner will be able to understand co-designed virtual machines and classic system virtual machines implementation |
| Unit V | The learner will be able to explore different types of system partitioning and emerging applications for virtual machine technology |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE106R01**

**Semester: I**

**ADVANCED DATABASE DESIGN**

**Course Objectives:**

To understand the advanced concepts in DBMS with case studies on Oracle and Microsoft SQL Server

**UNIT - I 12 Periods**

**Object and Object Relational Databases:** Overview of Object Database Concepts - Object-Relational Features: Object Database Extensions to SQL- The ODMG Object Model and the Object Definition Language ODL-Object Database Conceptual Design -The Object Query Language OQL-Overview of the C++ Language Binding in the ODMG Standard

**XML:** Structured, Semi-structured, and Unstructured Data- XML Hierarchical (Tree) Data Model-XML Documents, DTD, and XML Schema-Storing and Extracting XML Documents from Databases -XML Languages- Extracting XML Documents from Relational Databases

**UNIT - II 12 Periods**

**Distributed Databases:** Distributed Database Concepts-Types of Distributed Database Systems-Distributed Database Architectures - Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design-Query Processing and Optimization in Distributed Databases-Overview of Transaction Management in Distributed Databases- Overview of Concurrency Control and Recovery in Distributed Databases-Distributed Catalog Management-Current Trends in Distributed Databases -Distributed Databases in Oracle .

**UNIT - III 12 Periods**

**Parallel Databases:** Introduction-I/O Parallelism-Interquery Parallelism-Intraquery Parallelism-Intra-operation Parallelism-Interoperation Parallelism-Query optimization-Design of Parallel Systems-Parallelism on Multi-core Processors.

**Information Retrieval:** Overview-Relevance-Ranking Using Terms-Relevance Using Hyperlinks-Synonyms, Homonyms and Ontologies-Indexing of Documents-Measuring Retrieval Effectiveness-Crawling and Indexing the Web-Information Retrieval: Beyond Ranking of Pages-Directories and Categories.

**UNIT - IV 12 Periods**

**Enhanced Data Models for Advanced Applications:** Active Database Concepts and Triggers- Temporal Database Concepts-Spatial Database Concepts-Multimedia Database Concepts-Introduction to Deductive Databases

**UNIT - V 12 Periods**

**Case Studies: Oracle-**Database Design and Querying Tools-SQL Variations and Extensions-Storage and Indexing-Query Processing and Optimization-Concurrency Control and Recovery-System Architecture-Replication, Distribution and External Data-Database Administration Tools-Data Mining.

**Microsoft SQL Server**-Management, Design and Querying Tools-SQL Variations and Extensions-Storage and Indexing-Query Processing and Optimization-Concurrency and Recovery-System Architecture-Data Access-Distributed Heterogeneous Query Processing-Replication-XML Support-SQL Server Service Broker-Business Intelligence

**REFERENCES**

1. Abraham Silberschatz, Henry F. Korth, S.Sudarshan. *Database System Concepts.* McGraw-Hill International Edition, Sixth Edition, 2011.
2. Ramez Elmasri, Shamkant B Navathe. *Fundamentals of Database Systems.* Pearson Education, Sixth Edition, 2011.
3. Stefano Ceri, Giuseppe Pelagatti. *Distributed Databases Principles & Systems.* McGraw Hill International Editions, International Edition, 1985.
4. Tamer Ozsu, Patrick Valduriez. *Principles of Distributed Database Systems*. Springer, Third Edition, 2011.
5. Mario Piattini, Oscar Diaz. *Advanced Database Technology and Design.* Artech House, 2000.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand about objects and object relational Databases and XML |
| Unit II | The learner will be able to understand the main concepts of Distributed Databases |
| Unit III | The learner will be able to understand the concepts of Parallel Databases and Concepts of Information Retrieval |
| Unit IV | The learner will be able to get a quick overview of enhanced data models for Advanced Applications |
| Unit V | The learner will be able to make a case study on the two main databases, namely Oracle and SQL Server |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **0** | **0** | **4** | **2** |

**Course Code: MCSE107R02**

**Semester: I**

**PROGRAMMING LAB - I**

**Course Objectives:**

To help the learner to implement exercises based on machine learning, data structures, parallel computing and network sockets.

1. Write a menu driven program using ML to do the following.
2. A recursive function last to return the last element of a list.
3. A recursive function butLast to remove the last element of a list. For example, butLast[a,b,c,d] should return [a,b,c]. butLast(xs) should return [] if the list xs has length 1.
4. A function nth such that nth(xs,n) returns the nth element of list xs, counting the head of the list as element zero.
5. Write a menu driven program using ML to do the following.
6. Write a function startpoints(pairs,z)that produces the list of all x such that ( x; z) is in the list pairs. For example, startpoints ([(1,2), (2,3), (2,1)], 2) should yield [1].
7. Write a function endpoints(z,pairs) that produces the list of all y such that (z; y) is in the list pairs. For example, endpoints ([(1,2), (2,3), (2,1)], 2) should yield [3,1].
8. Write a function allpairs(xs,ys)that produces the list of all (x; y) for x in the list xs and y in the list ys.
9. Call a list of pairs complete if whenever (x; z) and (z;y)are in the list, then (x; y) is also in the list. (The empty list is trivially complete.) Write a function addnew((x,y),pairs), where you may assume the list pairs to be complete. The result should be a new list containing (x; y), the elements of pairs, and just enough additional pairs to make the result list complete. Concatenate lists generated with the help of the functions startpoints, endpoints, and allpairs.

3. (i) Write a ML code to generate the fifteenth fibbonacci number.

(ii) Write a function merge(xs,ys)that takes two increasing streams, x0< x1< x2< : : : and y0< y1< y2< : : :, and returns the increasing stream containing all the x's and y's.

(iii) Construct the increasing stream of all numbers of the form 2i3j 5k for integers i, j, k > 0. What is the sixtieth element of this stream. Write a ML code to implement this.

1. Create a dictionary of words as a trie structure and using that given a text file as input check whether all words in the file are correctly spelt or not.
2. For a given weighted connected undirected graph find the maximum flow and minimum cut.
3. Implement KMP pattern matching algorithm
4. Find the longest Palindrome substring from the given string using suffix trees.
5. Implement CUDA program to perform classical Gauss Elimination with back substitutions to solve linear equation.
6. Demonstrate simple Rule based mapping algorithm to adapt mapping between Virtual Machines in Cloud.
7. Implement CUDA program for database search using Smith-Waterman algorithm.
8. Create a chat application between two or more physical machines using inter process communication mechanism in UNIX socket programming environment.
9. Transfer a file between two or more physical machines using TCP with security in UNIX environment.
10. Transfer a file between two or more physical machines using UDP elementary sockets with security in UNIX environment.

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE201**

**Semester: II**

**STATISTICAL** **DATA ANALYTICS**

**Course Objectives:**

To understand regression models, perform simulation and use them in real world applications.

**UNIT - I 12 Periods**

**Introduction to R**: Vectors - Time series - Data Frames and Matrices - Functions, operators and Loops - Graphics - Styles of Data Analysis.

**UNIT - II 12 Periods**

**Statistical data models:** Statistical Models - Distribution models for the random component - Simulation of random numbers and samples - Model assumptions. Regression with a single predictor - Fitting a line to data - Outliers, influence, and robust regression - Standard errors and confidence intervals - Assessing predictive accuracy - Regression versus qualitative anova comparisons - issues of power - Logarithmic and other transformations - Model matrix in regression - Bayesian regression estimation.

**UNIT - III 12 Periods**

**Multiple Linear regression Models:** Multiple Linear regression - Introduction - Interpretation of model coefficients - Multiple regression assumptions, diagnostics, and efficacy measures - A strategy for fitting multiple regression models - Problems with many explanatory variables - Multicollinearity - Errors in x - Multiple regression models - Time series models.

**UNIT - IV 12 Periods**

**Mining Data**: Monte Carlo Simulations - Resampling Methods - Finding Clusters - Cluster - Distance and Similarity Measures - Clustering Methods - Pre- and Post processing - Sample applications Seeing the Forest for Trees : Finding Important Attributes - Principal Component Analysis - Visual Techniques - Kohonen Maps.

**UNIT - V 12 Periods**

**Applications using Data**: Reporting Business Intelligence and Dashboards - Business Intelligence - Corporate Metrics and Dashboards - Data Quality Issues. Financial Calculations and Modelling - The Time Value of Money - Uncertainty in Planning and Opportunity Costs - Cost Concepts and Depreciation - Problems in various applications.

**Predictive Analytics**: Introduction - Some Classification Terminology - Algorithms for Classification - The Process - The Secret Sauce - The Nature of Statistical Learning.

**REFERENCES**

1. John Maindonald, W. John Braun. *Data Analysis and Graphics Using R: An Example based Approach.* Cambridge University Press, Third Edition,2010.
2. Philipp K. Janert. *Data Analysis with Open Source Tools.* O'Reilly, 2010.
3. Andrew Gelman, Jennifer Hill. *Data Analysis using Regression and multilevel / Hierarchical Models.* Cambridge University Press, First Edition, 2006.
4. David Ruppert. *Statistics and Data Analysis for Financial Engineering.* Springer, 2011.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the features of R and be able to write simple programs using R |
| Unit II | The learner will be able to design models and perform regression techniques |
| Unit III | The learner will be able to build and apply linear regression models |
| Unit IV | The learner will be able to perform sampling apart from identifying ways of clustering in solving real world problems |
| Unit V | The learner will be able to apply the conceptual knowledge and use them in problems involving reporting and prediction |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE202**

**Semester: II**

**CLOUD COMPUTING**

**Course Objectives:**

To help the learner understand current cloud computing technologies, different services, cloud security and its applications.

**UNIT - I 12 Periods**

**Defining Cloud Computing**: Defining Cloud Computing - Cloud Types - Examining the Characteristics of Cloud Computing - Assessing the Role of Open Standards.

**Assessing the Value Proposition:** Measuring the Cloud's Value - Avoiding Capital Expenditures - Computing the Total Cost of Ownership - Specifying Service Level Agreements - Defining Licensing Models.

**Understanding Cloud Architecture:** Exploring the Cloud Computing - Connecting to the Cloud.

**UNIT - II 12 Periods**

**Understanding Services and Applications by Type:** Defining Infrastructure as a Service (IaaS) - Defining Platform as a Service (PaaS) - Defining Software as a Service (SaaS) - SaaS characteristics- Defining Identity as a Service (IDaaS) - Defining Compliance as a Service (CaaS).

**Understanding Abstraction and Virtualization:** Using Virtualization Technologies - Load Balancing and Virtualization - Understanding Hypervisors - Understanding Machine Imaging - Porting Applications.

**Exploring Platform as a Service:** Defining Services - Using PaaS Application Frameworks.

**UNIT - III 12 Periods**

**Using Google Web Services:** Exploring Google Applications - Surveying the Google Application Portfolio - Exploring the Google Toolkit - Working with the Google App Engine. **Using Amazon Web Services:** Understanding Amazon Web Services- Amazon Web Service Components and Services - Working with the Elastic Compute Cloud (EC2) - Working with Amazon Storage Systems - Understanding Amazon Database Services.

**Using Microsoft Cloud Services:** Exploring Microsoft Cloud Services - Defining the Windows Azure Platform - Using Windows Live.

**UNIT - IV 12 Periods**

**Understanding Cloud Security:** Securing the Cloud - Securing Data - Establishing Identity and Presence.

**Understanding Service Oriented Architecture:** Introducing Service Oriented Architecture - Defining SOA Communications - Managing and Monitoring SOA - Relating SOA and Cloud Computing.

**Moving Applications to the Cloud**: Applications in the Clouds - Applications and Cloud APIs. **Working with Cloud-Based Storage:** Measuring the Digital Universe Provisioning Cloud Storage - Exploring Cloud Backup Solutions - Cloud Storage Interoperability.

**UNIT - V 12 Periods**

**Understanding Scientific Applications for Cloud Environments:** Classification of Scientific Applications and Services in the Cloud - SAGA-based Scientific Applications that Utilize Clouds. **MapReduce Programming Model and Implementations:** MapReduce Programming Model -Major MapReduce Implementations for the Cloud - MapReduce Impacts and Research Directions.

**An Architecture for Federated Cloud Computing:**  A Typical Use Case- The Basic Principles of Cloud Computing- A Model for Federated Cloud Computing - Security.

**SLA Management in Cloud Computing:** Inspiration - Traditional Approaches to SLO Management -Types of SLA - Life Cycle of SLA- SLA Management in Cloud - Automated Policy-based Management.

**REFERENCES**

1. Barrie Sosinsky. *Cloud Computing Bible.* Wiley Publishing, First Edition, 2011.
2. Rajkumar Buyya, James Broberg, Andrzej Goscinski. *Cloud Computing Principles and Paradigms.* John Wiley & Sons, 2011.
3. Dan Sullivan. *The Definitive Guide to Cloud Computing.* Realtime publishers, 2010.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the basics of cloud computing, its pros & cons and the cloud architecture |
| Unit II | The learner will be able to acquire knowledge on different services and the virtualization technologies |
| Unit III | The learner will be able to understand services provided by different providers like Google, Amazon and Microsoft |
| Unit IV | The learner will be able to understand service models and their security levels, standards used to create cloud service and cloud storage |
| Unit V | The learner will be able to understand scientific applications for cloud environment, MapReduce programming model and SLA management |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **3** | **1** | **0** | **4** |

**Course Code: MCSE203**

**Semester: II**

**SOFTWARE TESTING & AUTOMATION**

**Course Objectives:**

To help the learners understand the testing artifacts, to develop software test plan, to design effective test cases and to learn the automated software testing tools.

**UNIT - I 12 Periods**

Testing Fundamentals - Defects - Strategies and Methods for Black Box Test Case Design - Strategies and Methods for White-Box Test Case design -Levels of Testing and Different Types of testing - System Testing - types of system testing - Acceptance testing - Performance testing - Regression Testing -Alpha - Beta Tests - testing OO systems - usability and accessibility testing.

**UNIT - II 12 Periods**

Organization and Development of Testing Approach Overview of the software testing process - Organizing for testing - Developing Test plan - Profile the software project - Understand project risk - Testing technique - UNIT testing and analysis - Build and Inspect Test Plan.

**UNIT - III 12 Periods**

Verification and Validation Verification Testing - Requirement phase Testing - Design phase testing - Programming phase testing - Test during requirement, Design and Programming Phase - Guidelines - Validation Testing - Build test data - Execute Results - Record Test Results.

**UNIT - IV 12 Periods**

Implementation Acceptance Testing and Operational Testing - Acceptance Testing - Define, Develop and Execute - Preoperational Testing - Test and Monitor - Post-Operational Testing - Develop and Test - Post Implementation Analysis - Workbenches - Procedures.

**UNIT - V 12 Periods**

Introduction to automated Testing - Test Automation for Web Applications - Introduction to Selenium-Getting started with Selenium IDE - Locators-Overview of Selenium WebDriver - Design Patterns - Finding Elements - Working with WebDriver - Getting started with Selenium Grid.

**REFERENCES**

1. William E Perry. *Effective Methods for Software Testing.* John Wiley & Sons, Third Edition, 2006.
2. Ilene Burnstein. *Practical Software Testing.* Springer International Edition, Chennai, 2003.
3. Srinivasan Desikan, Gopalaswamy Ramesh. *Software Testing - Principles and Practices.* Pearson education, 2006.
4. David Burns. *Selenium 2 Testing Tools - Beginners Guide.* Packt Publishing, 2012.

**ONLINE MATERIALS**

1. [www.seleniumhq.org](http://www.seleniumhq.org).

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to have an understanding of the fundamental concepts in software testing strategies |
| Unit II | The learner will be able to build and inspect a test plan |
| Unit III | The learner will be able to develop test requirements, test scenarios, test cases for performing Verification and Validation |
| Unit IV | The learner will be able to define, develop and execute acceptance and operational testing |
| Unit V | The learner will be able to analyze and execute the automated software testing process using the tools like Selenium |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **3** | **0** | **0** | **3** |

**Course Code: MCSE204**

**Semester: II**

**WEB SERVICES**

**Course Objectives:**

To provide an in-depth knowledge of Service Oriented Architecture and its various messaging primitives and the technology, models and protocols of web services.

**UNIT - I 9 Periods**

**Introducing SOA:** Fundamental SOA - Common characteristics of contemporary SOA - Common misperceptions about SOA - Common tangible benefits of SOA - Common pitfalls of adopting SOA

**Evolution of SOA:** An SOA timeline - Web services: a brief history - The roots of SOA

**UNIT - II 9 Periods**

**Activity Management and Composition:** The Web services framework - Services - Service descriptions (with WSDL) - Messaging (with SOAP).

**SOA and WS-\* Extensions:** Introducing WS-\* - Message exchange patterns - Service activity - Coordination - Atomic transactions - Business activities - Orchestration - Choreography.

**UNIT - III 9 Periods**

**Advanced Messaging, Metadata, and Security:** Addressing - Reliable messaging - Correlation - Policies - Metadata exchange - Security - Notification and eventing.

**Principles of Service-Orientation:** Service-orientation and the enterprise - Anatomy of a service-oriented architecture - Common principles of service-orientation.

**UNIT - IV 9 Periods**

**Web Services:** Web Services and their Approach to Distributed Computing - Web Services Technologies - Web Services Architecture.

**Basic Web Services Technology:** A Minimalist Infrastructure for Web Services - SOAP - WSDL - UDDI - Web Services at Work - Interactions Between the Specifications - Related Standards.

**UNIT - V 9 Periods**

**Service coordination protocols:** An Introduction to Coordination Protocols - Infrastructure for Coordination Protocols - WS-coordination - WS-Transaction - RosettaNet - Other Standards.

**Service Composition:** Basics of Service Composition - Service Composition Models.

**REFERENCES**

1. Thomas Erl. *Service-Oriented Architecture: Concepts, Technology and Design*, Pearson Education, First Edition, 2005.
2. Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju. *Web Services: Concepts, Architectures and Applications*, Springer 2006.
3. Michael Papazoglou. *Web Services and SOA: Principles and Technology*, Pearson Education, Second Edition, 2012.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to acquire idea about the basics of SOA and its benefits |
| Unit II | The learner will be able to understand the framework of web services and the various primitives for exchange of messages |
| Unit III | The learner will be able to understand the advanced level issues in messaging and security policies |
| Unit IV | The learner will be able to understand the architecture of web services and the basic technologies behind it |
| Unit V | The learner will be able to understand the need for coordination among services the protocols behind it |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **0** | **0** | **4** | **2** |

**Course Code: MCSE207**

**Semester: II**

**PROGRAMMING LAB - II**

**Course Objectives:**

To help the learner to implement exercises based on statistical data models, cloud clusters, web services and software testing.

1. Perform regression analyses and anova testing on a set of thousand data.

(A suitable set of benchmark data can be collected from online datasets.)

1. Simulate
2. Delta modulation  b. Adaptive delta modulation. Use either matlab / scilab for performing the same.
3. Consider the example of filling petrol from the bunk. It could be observed that people arrive from time to time. ( There may be cases when no person is available in the bunk ). Consider server idle and waiting times. Simulate using matlab in combination with simulink taking the following two cases into consideration.

        a. Single server queues b. Multi Server queues

1. Creating an interactive Hadoop Map reduce job flow using JCUDA.
2. Implement group communication between multiple GPU’s.
3. Implement CUDA program to validate any one of the hash algorithm by comparing VM’s and PM’s performance.
4. Invoking a windows applications through a web service .
5. Create addition web service in ASP.Net invoke it in using C#.
6. Discovering and Invoking Web Services USING RESTful API.
7. Develop selenium test cases to Explore verify and wait commands with parameters pre-defined according to the context of the selected UI element.
8. Develop selenium test cases to Explore store, alert and confirmation commands with parameters pre-defined according to the context of the selected UI element.
9. Implement selenium script to support dynamic web pages to support modern advanced web-app testing problems.
10. Write a selenium script to run against different browsers in parallel.

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: MCSE301**

**Semester: III**

**MULTI AGENT SYSTEMS**

**Course Objectives:**

To understand the basic principles of the Multi-agent Systems operation main techniques and to know how to use them in the environment of an intelligent service or system.

**UNIT - I 12 Periods**

MultiAgent problem Formulation - Utility, Markov Decision Process, Hierarchical Planning Distributed Constraint Satisfaction - Domain Pruning Algorithms, Heuristic Search. Algorithms -Distributed Optimization - Action selection in multiagent MDP’s - Negotiations, Auctions and Optimization - Distributed Constraint Optimization - Adopt, OptAPO.

**UNIT - II 12 Periods**

Self interested Agents - Games in Normal Form - Analyzing games - Computing solution concepts of normal form games - Games with sequential actions - Perfect and Imperfect Information Extensive form games.

**UNIT - III 12 Periods**

Richer representation - Beyond the Normal and extensive forms - Repeated, Stochastic - Bayesian - Congestion Games - Computationally motivated Compact representations - Terms of Selfish Agents - Coalition Games with transferable utility - Analysis - Compact representations.

**UNIT - IV 12 Periods**

Learning and Teaching - Introduction - Fictitious Play - Rational Learning - Reinforcement

Learning - No Regret Learning - Targeted Learning - Evolutionary Learning and Large Population Models - Communication. Protocols for Strategic Agents - Mechanism Design with unrestricted preferences - Quasilinear preferences.

**UNIT - V 12 Periods**

Logics of Knowledge and Belief - Partition Model of Knowledge - Modal Logic - Axiomatic theory of partition model - Common knowledge and related applications - Knowledge to Belief - Combining Knowledge and Belief. Beyond Belief - Knowledge and Probability - Dynamics of Knowledge and Belief - Logic, Games and Coalition Logic - Towards a logic of intention.

**REFERENCES**

1. Yoav Shoham, Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2011.
2. Jos´e M Vidal. *Fundamentals of Multiagent Systems with NetLogo Examples.* Kevin Leyton-Brown, 2010.
3. Richard Murch, Tony Johnson. *Intelligent Software Agents.* Prentice Hall, 2000.
4. Michael Woodridge. *An Introduction to Multiagent Systems.* John Wiley & Sons Ltd., Second Edition, 2009.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to formulate problems and deal with suitable methods of optimization |
| Unit II | The learner will be able to understand the working of conflicting agents and use the knowledge in identifying solutions |
| Unit III | The learner will be able to make appropriate representation for various stochastic problems |
| Unit IV | The learner will be able to identify the kind of learning technique to be used in solving real world problems |
| Unit V | The learner will be able to design a belief network for a real world problem |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E01**

**Semester:**

**MODERN EMBEDDED COMPUTING**

**Course Objectives:**

To help the learner to understand the platform architecture of today’s embedded computing systems.

**UNIT - I 12 Periods**

**Future of Embedded Systems:** Technology Trends - Issues, Applications, Initiatives - Challenges and Uncertainties.

**Embedded Platform Architecture:** Platform Overview - Volatile Memory Technologies - Non volatile Storage - Device Interface - High Performance Universal Serial Bus - Device Interconnect - Low Performance - General Purpose Input/Output - Power Delivery.

**UNIT - II 12 Periods**

**Embedded Processor Architecture:** Basic Execution Environment - Application Binary Interface - Processor Instruction Classes - Exceptions/Interrupts Model - Vector Table structure - Exception frame - Mashing & Acknowledging Interrupts - Interrupt latency - Memory Mapping and Protection - MMU & Process - Memory Hierarchy - Intel Atom Microarchitecture - Embedded Platform Boot Sequence.

**UNIT - III 12 Periods**

**Operating Systems Overview -** Application Interface - Processes - Tasks and Threads - Scheduling - Memory - Clocks and Timers - Mutual Exclusion/Synchronization - Device Driver Models - Bus Drivers - Networking - Storage File Systems - Power Management - Real Time - Power Optimization-

**Embedded Graphics and Multimedia Acceleration:** Screen Display - Embedded Panels - Graphics Stack - Accelerated Media Decode - Video Capture and Encoding - Media Frameworks.

**UNIT - IV 12 Periods**

**Digital Signal Processing Using General - Purpose Processors:** Overview - Single Instruction Multiple Data - Microarchitecture Considerations - Implementation Options - Intrinsics and data types - Finite Impulse Response Filter - Application Examples.

**Network Connectivity:** Networking Basics - ТСР/IP Networking - Ethernet - Wi-Fi and IEEE 802.11 365 - Bluetooth - Linux Networking.

**UNIT - V 12 Periods**

**Platform and Content Security:** Security Principles - Security Concepts and Building Blocks - Platform Support for Security - Multiprocessing - Symmetric Multiprocessing - Asymmetric Multiprocessing.

**Virtualization:** Basics - Methods for Platform Virtualization.

**Performance Tuning:** General approaches - Code and Design - Processor-Specific - Networking Techniques.

**REFERENCES**

1. Peter Barry and Patrick Crowley**.**  *Modern Embedded Computing Designing Connected, Pervasive*. Media-Rich Systems. Elsevier Morgan Kaufmann, First Edition, 2012.
2. Andrew N Sloss, Dominic Symes, Chris Wright. *ARM System Developers Guide designing and optimizing system software*. Elsevier, 2004.
3. Raj Kamal. *Embedded systems - Architecture, Programming and design.* Tata McGraw Hill, Fifth Reprint, 2010.
4. Wayne Wolf. *Computers as Components - Principles of Embedded Computing System Design.* Elsevier, Second Edition, 2008.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to know about future of embedded systems and its architecture |
| Unit II | The learner will be able to have an understanding about the Embedded execution environment |
| Unit III | The learner will be able to know about the operating system overview and embedded graphics and multimedia |
| Unit IV | The learner will be able to know about Digital Signal Processing Using General Purpose Processors |
| Unit V | The learner will be able to know about security and performance tuning |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E02**

**Semester:**

**COMPUTER VISION & MACHINE LEARNING**

**Course Objectives:**

To facilitate the learner in providing computers with human like perception capabilities to sense environment, understand data and learn from experience to enhance future performance

**UNIT - I 12 Periods**

Throwing down the Visual Intelligence Gauntlet - Neuromorphic approach - Quest for Visual Intelligence - Actionable Information in Vision -Probabilistic Classifiers - Maximum likelihood classification - Information Theory - Inequalities - Bayes optimal error and Entropy - Analysis of classification error of estimated distribution - Density of distributions - Complex probabilistic models

**UNIT - II 12 Periods**

Generalized Bounds - Margin distribution based Bound - Analysis-Semi supervised Learning - Properties of Classification - Semi-supervised Learning using Maximum likelihood estimation - Asymptotic properties - Learning with finite data -Maximum Likelihood Minimum Entropy - Mutual Information, Bayes Optimal Error, Entropy and Conditional Probability - Maximum Mutual Information HMMs - Convexity - Convergence

**UNIT - III 12 Periods**

Margin Distribution Optimization - Base Bound - Existing Learning algorithms - Margin+ distribution Optimization algorithm - Learning the structure of Bayesian Classifiers - Network Classifiers - Stochastic search algorithm - Active learning-Learning Binary Hash codes for Large Scale Image search - Search algorithms - Supervised Method for learning Binary projections - Unsupervised Method for defining Binary projections

**UNIT - IV 12 Periods**

Bayesian Painting by Numbers : Flexible Priors for Colour Invariant Object recognition - Colour Invariant Admixture Model - Using Steps for Supervised Learning Tasks -Real Time Human Pose Recognition in Parts from single Depth Images - Data - Body Part Inference and Joint Proposals - Identifying accuracies-Scale Invariant Vote Based 3D recognition and Registration from Point Clouds

**UNIT - V 12 Periods**

Multiple Classifier Boosting and Tree structured Classifiers - Multiple classifier Boosting - Boosting for Object Tracking - Conversion to Decision tree by Boolean optimization -Simultaneous Detection and Tracking with Multiple Cameras - Applications of Computer Vision to Vehicles: An Extreme Test - Motivation - Sensors - Processing Systems

**REFERENCES**

1. Roberto Cipolla, Sebastiano Battiato, Giovanni Maria Farinella. *Machine Learning for Computer Vision*. Studies in Computational Intelligence, Springer, 2013.
2. [N. Sebe](http://link.springer.com/search?facet-author=%22N.+Sebe%22),  [Ira Cohen](http://link.springer.com/search?facet-author=%22Ira+Cohen%22), [Ashutosh Garg](http://link.springer.com/search?facet-author=%22Ashutosh+Garg%22), [Thomas S. Huang](http://link.springer.com/search?facet-author=%22Thomas+S.+Huang%22). *Machine Learning in Computer Vision.* Vol.29., Springer, 2005.
3. Simon J D Prince. *Computer Vision: Models, Learning, and Inference.* Cambridge University Press, First Edition, 2012.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to produce truly intelligent machines dealing with aspects of vision |
| Unit II | The learner will be able to apply probability and entropy concepts in performing classification operations |
| Unit III | The learner will be able to design algorithms for rapidly search images or videos in large collections |
| Unit IV | The learner will be able to design methods for quickly and accurately predict 3D positions of body joints from a single depth image |
| Unit V | The learner will be able to design algorithms for object detection, tracking and segmentation |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E03**

**Semester:**

**WIRELESS SENSOR NETWORKS**

**Course Objectives::**

The learner will understand different sensor node technology, routing design and the requirements of operating system to deploy WSN.

**UNIT - I 12 Periods**

Introduction and Overview of Wireless Sensor Networks- Basic Sensor Network Architectural Elements- Challenges and Hurdles, Examples of category 1 and 2 WSN application, Sensor node technology- Hardware and Software- Wireless Node Operating environment, Wireless Transmission Technology and System- Available Wireless Technologies, Medium Access Control Protocols for Wireless Sensor Network- Fundamentals of MAC protocols- MAC protocols for WSNs.

**UNIT - II 12 Periods**

Routing Protocols for Wireless Sensor Networks- Routing Challenges and Design Issues in Wireless Sensor Networks, Routing Strategies in Wireless Sensor Network, Network Layer - Data-centric and Flat-Architecture Protocols, Hierarchical Protocol-LEACH-PEGASIS, Geographical Routing Protocols- MECHC and SMECN-PRADA, Transport Layer-Challenges for Transport layer-RMST -CODA-ESRT- Real-Time and Reliable Transport Protocol (RT)2 protocol.

**UNIT - III 12 Periods**

Node Clustering and Security in WSN-Node Clustering structures-Node Clustering Algorithms-Cluster-Head Election Algorithms-Node Clustering algorithms for WSNs, Query processing and Data Aggregation-query processing in WSNs-Data Aggregation in WSNs, Network Security and Attack Defense-confidentiality-integrity-authenticity-nonrepudiation-availability-Intrusion Detection-key management.

**UNIT - IV 12 Periods**

Network management and operating system- Network management requirements, Traditional network management models, Network management design issues, MANNA, Operating system-design issues, TinyOS, Mate, MagnetOS, MANTIS, OSPM, EYES OS, SenOS, EMERALDS, PicOS. Performance and Traffic Management -WSN design issues-performance modelling of WSNs Case study: Simple computation of the System Life Span.

**UNIT - V 12 Periods**

Topology Control - Distributed Topology Control- Design Guidelines -Ideal Features of a Topology Control Protocol. Future Trends in WSNs-Wireless Multimedia Sensor Networks-Wireless Sensor and Actor Networks -sensor network applications in Challenging Environments-underwater acoustic sensor networks-wireless underground sensor networks.

**REFERENCES**

1. Kazem Sohraby, Daniel Minoli, Taieb Znati. *Wireless Sensor Networks Technology - Protocols and Applications*. John Wiley & Sons, Ltd.,2007.
2. Jun Zheng, Abbas Jamalipour. *Wireless Sensor Networks: A Networking Perspective*. John Wiley & Sons, Ltd., 2009.
3. Ian F. Akyildiz. *Wireless Sensor Networks.* John Wiley & Sons, Ltd., 2010.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to have knowledge about the overview of sensor architecture for WSN |
| Unit II | The learner will be able to have knowledge about the design and challenges of Routing protocol in network layer, protocol design for Transport layer |
| Unit III | The learner will be able to have knowledge about the node selection, algorithm used for node clustering and learn about the security issues |
| Unit IV | The learner will be able to have knowledge about the network management and Operating system used for WSNs |
| Unit V | The learner will be able to have knowledge about the design guideline of topology and about the future trends available in WSNs |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E04**

**Semester:**

**FAULT TOLERANT COMPUTING**

**Course Objectives:**

To enable the learners understand various techniques and approaches used in Fault Tolerance in computer systems. To gain knowledge in sources of faults and means for their prevention and to understand merits and limitations of fault-tolerant design

**UNIT - I 12 Periods**

**Introduction:** Fault Classification -Types of Redundancy - Basic Measures of Fault Tolerance. **Hardware Fault Tolerance :**The Rate of Hardware Failures - Failure Rate, Reliability, and Mean Time to Failure - Canonical and Resilient Structures - Other Reliability Evaluation Techniques -Fault-Tolerance Processor-Level Techniques - Byzantine Failures.

**Information Redundancy:** Coding - Resilient Disk Systems.

**UNIT - II 12 Periods**

**Fault-Tolerant Networks:** Measures of Resilience - Common Network Topologies and Their Resilience- Fault-Tolerant Routing.

**Software Fault Tolerance:** Acceptance Tests - Single-Version Fault Tolerance -N-Version Programming - Recovery Block Approach - Preconditions, Post conditions, and Assertions - Exception-Handling - Software Reliability Models - Fault-Tolerant Remote Procedure Calls.

**UNIT - III 12 Periods**

**Checkpointing:** Check pointing- Checkpoint Level - Optimal Checkpointing—An Analytical Model - Cache-Aided Rollback Error Recovery (CARER) - Checkpointing in Distributed Systems - Checkpointing in Shared-Memory Systems - Other Uses of Checkpointing.

**Fault Detection in Cryptographic Systems:** Overview of Ciphers - Security Attacks Through Fault Injection – Countermeasures.

**Simulation Techniques:** Parameter Estimation - Variance Reduction Methods - Fault Injection.

**UNIT - IV 12 Periods**

**Nonstop Systems-** Stratus Systems - Cassini Command and Data Subsystem - IBM G5 - IBM Sysplex – Itanium.

**Defect Tolerance in VLSI Circuits -**Manufacturing Defects and Circuit - Probability of Failure and Critical Area - Basic Yield Models - Yield Enhancement through.

**Programs for Reliability Modeling and Analysis:** Introduction-Various Types of Reliability and Availability Programs-Testing Programs- Partial List of Reliability and Availability Programs-An Example of Computer Analysis.

**UNIT - V 12 Periods**

**Software Reliability and Recovery Techniques:** Introduction-The Magnitude of the Problem- Software Development Life Cycle- Reliability Theory- Software Error Models- Reliability Models- Estimating the Model Constants-Other Software Reliability Models-Software Redundancy-Rollback and Recovery

**REFERENCES**

1. Israel Koren, C. Mani Krishna. *Fault-Tolerant Systems.* Elsevier, 2007.
2. M. L. Shooman. *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis and Design.* Wiley Interscience, 2002.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to have knowledge about the basic classification of the faults, hardware failures and basics about the reliability |
| Unit II | The learner will be able to have clear idea about the fault tolerant networks and software faults and Tests |
| Unit III | The learner will be able to understand checkpointing, detecting faults using cryptographic techniques and simulation techniques |
| Unit IV | The learner will be able to understand of VLSI circuit tolerance and analysis and modelling of program for reliability |
| Unit V | The learner will be able to understand the techniques for failure recovery and methods to improve software reliability |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E05**

**Semester:**

**IMAGE PROCESSING & PATTERN RECOGNITION**

**Course Objectives:**

To enable the learners to acquire knowledge on the various components of image processing like image acquisition, enhancement and restoration in spatial and frequency domain, segmentation, representation and description. To understand the representation, feature extraction and feature selection methods for pattern recognition. To adapt the various classifiers to pattern recognition problems

**UNIT - I 12 Periods**

**Fundamentals of Image Processing**: Digital image fundamentals, Components of Image Processing System, Elements of visual perception, Sensing and Acquisition, Sampling and Quantization matrix, Relationship between Pixels, Linear and Non-linear Operations, Discrete Fourier transform & Wavelet transform.

**UNIT - II 12 Periods**

**Image Enhancement:** Point operations - Basic gray level transformations, Histogram Processing, Enhancement using Arithmetic/logic Operations, Spatial Filters: Basics, Smoothing filters, Sharpening Filters. Frequency Domain: Smoothing, Sharpening filters, Homomorphic Filtering. Image Restoration: Noise models, Mean filters, Order statistic filters, Adaptive filters, Inverse Filtering, Wiener Filtering, Geometric Mean Filter.

**UNIT - III 12 Periods**

**Image Segmentation:** Point, Line and Edge Detection, Thresholding, Region - Based Segmentation, Use of Motion in Segmentation. Image Representation and Description: Representation, Boundary and Regional Descriptors, Principal Component for Description, Relational Descriptor.

**UNIT - IV 12 Periods**

**Pattern Representation:** Data Structures for Pattern Representation, Representation of Clusters, Proximity Measures, Size of Patterns, Abstraction of Data Set, Feature Extraction, Feature Selection, Evaluation of Classifiers, Evaluation of Clustering. Nearest Neighbor based Classifiers: NN Algorithm, Variants of NN, Efficient Algorithms, Data Reduction, Prototype Selection.

**UNIT - V 12 Periods**

**Bayes Classifier**: Bayes Theorem, Minimum Error Rate Classifier, Estimation of Probabilities, Naïve Bayes Classifier, Bayesian Belief Network. Hidden Markov Models: Markov Models for Classification, HMM, Classification using HMM. Decision Trees: Introduction, Decision Trees for Classification, Construction of Decision Trees, Splitting at the Nodes, Over-fitting and Pruning. Support Vector Machines, Learning the Linear Discriminant Function, SVM for Classification.

**REFERENCES**

1. Rafael C.Gonzalez, Richard E.Woods. *Digital Image Processing.* Pearson Education, Third Edition, 2008.
2. M. Narasimha Murty, V.Susheela Devi. *Pattern Recognition - An Algorithmic Approach.* University Press, Springer, 2011.
3. S. Sridhar. *Digital Image Processing*. Oxford University Press, 2011.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learners will be able to comprehend image acquisition, sampling and quantization, image transforms and their properties. |
| Unit II | The learners will be able to enhance and restore images in the spatial and frequency domains. |
| Unit III | The learners will be able to acquire the knowledge to segment and represent images. |
| Unit IV | The learners will be able to understand the basics of pattern recognition like pattern representation, abstraction, feature extraction and selection and classification using NN Classifier. |
| Unit V | The learners will be able to classify objects using Naïve Bayes, HMM, Decision trees and SVM. |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E06**

**Semester:**

**MOBILE & UBIQUITOUS COMPUTING**

**Course Objectives:**

To understand the characteristics of wireless mobile communication and various protocols used in different layers of ad hoc networks.

**UNIT - I 12 Periods**

**Introduction to Ad Hoc networks**: Introduction - characteristics of Wireless channel - differences between cellular and ad hoc wireless networks - applications of ad hoc wireless networks - issues in ad hoc wireless networks - Standards - IEEE 802.11 - HIPERLAN - Bluetooth.

**UNIT - II 12 Periods**

**MAC Protocols**: Design issues - design goals - classification of MAC protocols - contention based protocols - MACAW - FAMA - BTMA - MACA-BI - MARCH - contention based protocols with reservation - D-PRMA - CATA - HRMA - SRMA/PA - FPRP - MACA/PR - RT-MAC - contention based protocols with scheduling algorithms - DPSMA - DWOP - DLPS - protocols using directional antennas - MMAC - MCSMA - PCMAC - RBAP - ICSMA.

**UNIT - III 12 Periods**

**Routing Protocols**: Design issues, goals and classification - table-driven protocols - DSDV - WRP - CGSR - STAR - on demand protocols - DSR - AODV - TORA - LAR - ABR - SAR - FORP - hybrid protocols - CEDAR - ZRP - Z-HLSR - efficient routing with flooding mechanism - PLBR - OLSR - hierarchical routing - HSR - FSR - power aware routing.

**UNIT - IV 12 Periods**

**Multicast routing protocols**: issues, operation, architecture and classification - tree-based multicast routing - mesh-based multicast routing - energy efficient multicast routing - multicasting with QoS guarantees - application dependant multicasting.

**UNIT - V 12 Periods**

**Transport layer:** design issues, goals and classification - Traditional TCP - TCP-F - TCP-ELFN - TCP-BuS - ad hoc TCP - split TCP

**Security:** Issues and challenges, network security attacks, key management - secure routing protocols.

**REFERENCES**

1. C. Siva Ram Murthy, B. S. Manoj. *Ad hoc Wireless Networks Architectures and protocols*. Pearson Education, Second Edition, 2012.
2. Subir Kumar Sarkar, T.G. Basavaraju, C. Puttamadappa. *Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications.* CRC Press, Second Edition, 2013.
3. Charles E. Perkins. *Ad hoc Networking.* Addison Wesley, First Edition, 2000.
4. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan Stojmenovic. *Mobile Ad hoc Networking.* Wiley-IEEE press, 2004.
5. Chai K Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall, First Edition, 2001.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the characteristics of wireless communication, issues in ad hoc network design and the wireless communication standards |
| Unit II | The learner will be able to explore the different kinds MAC protocols designed for ad hoc networks |
| Unit III | The learner will be able to understand the various routing protocols designed for mobile networks |
| Unit IV | The learner will be able to understand the various multicast routing protocols designed for mobile networks |
| Unit V | The learner will be able to explore the various transport protocols and security protocols designed for mobile networks |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E07**

**Semester:**

**SOFTWARE ARCHITECTURE & DESIGN PATTERNS**

**Course Objectives:**

The learner will understand the role of software architecture and different patterns and will learn about how it is related to other portions of software life cycle along with its desired quality attributes.

**UNIT - I 12 Periods**

**Envisioning Architecture:** The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views.

**Creating an Architecture**: Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

**UNIT - II 12 Periods**

**Analyzing Architectures:** Architecture Evaluation, Architecture design, decision making, ATAM, CBAM. Moving from one system to many: Software Product Lines, Building systems from off the shelf components, Software architecture in future - Introduction to open architecture standards.

**UNIT - III 12 Periods**

**Creational pattern:** Abstract Factory-Builder-Factory method

**Structural patterns:** Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight, Proxy.

**UNIT - IV 12 Periods**

**Behavioral patterns:** Chain of responsibility, command, Interpreter, Iterator, mediator, memento, observer, state, strategy, template method, visitor.

**UNIT - V 12 Periods**

**Case Studies:** A-7E - A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control - a case study in designing for high availability, Celsius Tech - a case study in product line development.

**REFERENCES**

* 1. Len Bass, Paul Clements, Rick Kazman. *Software Architecture in Practice.* Pearson Education, Second Edition, 2003.
  2. Richard N. Taylor, Nenad Medvidovic, Eric Dashofy. *Software Architecture: Foundations.* Theory and Practice, Addison Wesley, 2009.
  3. Erich Gamma. *Design Patterns.* Pearson Education, First Edition, 1995.

1. Luke Hohmann. *Beyond Software Architecture*, Addison Wesley, 2003.
2. David M. Dikel, David Kane, James R. Wilson. *Software Architecture*. Prentice Hall, First Edition, 2001.
3. F. Buschmann. *Pattern Oriented Software Architecture*. Wiley & Sons, First Edition, 2001.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to design software architecture with the desired quality attributes |
| Unit II | The learner will be able to implement and integrate it with other portions of software development life cycle |
| Unit III | The learner will be able to choose an appropriate design pattern to solve design problems |
| Unit IV | The learner will be able to learn how these important patterns fit into the software development process and the way they can leverage them to solve your own design problems most efficiently |
| Unit V | The learner will be able to understand the real time application of the different architectures |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E08**

**Semester:**

**SEMANTIC WEB & SOCIAL NETWORKS**

**Course Objectives:**

To describe the various concepts and technologies that make up the Semantic Web landscape, Prepare artifacts (e.g. component ontologies, architectures) and assess the value and applicability of Semantic Web approaches to various problems.

**UNIT - I 12 Periods**

**RDF Introduction**: Towards the idea of Semantic web - Building Block of Semantic Web - Abstract Model of RDF - RDF Serialization - Formats - Rules of RDF - XML vs RDF - Use of a RDF Validator.

**UNIT - II 12 Periods**

**RDF related technologies**: Micro formats - RDFa - GRDDL .RDFS and Ontology - Core Elements - Concept of Ontology - Building Bridge to Ontology - Inferencing based on RDF schema.

**UNIT - III 12 Periods**

**OWL :** OWL1 Web ontology Languages - Defining Classes, Properties - Camera Ontology OWL2 Web ontology Languages - New Constructs for Common Patterns - Improved Expressiveness for Properties - Extended Support for Data types - Punning and Annotations - Other OWL 2 Features - OWL Constructs in Instance Documents v- OWL 2 Profiles - Camera Ontology.

**UNIT - IV 12 Periods**

**Managing Space and Time**: Space and Time in Software - Spatial Information - Temporal Information - Representing Spatiotemporal Data on the Semantic Web - Working with Spatial Data - Spatial and Transaction Time Bounded Queries Framing the Problem - Approach and Rationale - Components. Aggregating Disparate Data Sources - Annotating Unstructured Data - Annotation Management.

**UNIT - V 12 Periods**

**Social Networks and the Semantic Web** **:** Overview of Social Networking Websites -Facebook’s Open Graph Protocol - Open Graph Protocol - How Does It Work: Creating Typed Links Using OGP - Implications for the Semantic Web -Twitter Cards for Structured Information - Rich Pins for Structured Information .Building the Foundation for Development on the Semantic Web - Development Tools for the Semantic Web - Frameworks for the Semantic Web Applications - Reasoners for the Semantic Web Applications - Ontology Engineering Environments - **Other Tools**: Search Engines for the Semantic Web - Semantic Web Application Development Methodology - From Domain Models to Ontology-Driven Architecture - An Ontology Development Methodology Proposed by Noy and McGuinness

**REFERENCES**

|  |
| --- |
|  |

1. Liyang Yu. *A Developer’s Guide to the Semantic Web.* Springer, Second Edition, 2014.
2. John Hebeler, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez. *Semantic Web Programming.* Wiley India, First edition, 2009.
3. John Davies, Rudi Studer, Paul Warren. *Semantic Web Technologies: Trends and Research in Ontology-Based Systems.* Wiley India, 2006.
4. Grigoris Antoniou, Frank Van Harmelen. *A Semantic Web Primer, MIT Press Cooperative Information Systems.* Second Edition, 2008.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to provide a comprehensive exposition of the state-of-the art in Semantic Web research and key technologies |
| Unit II | The learner will be able to explain the use of ontologies and metadata to achieve machine-interpretability |
| Unit III | The learner will be able to describe methods for ontology learning and metadata generation |
| Unit IV | The learner will be able to identify and apply various tools and techniques for working with spatial data |
| Unit V | The learner will be able to illustrate the theoretical concepts with case studies on industrial applications |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E09**

**Semester:**

**BIG DATA ANALYTICS**

**Course Objectives:**

This course covers foundational techniques and tools required for data science and big data analytics. The course focuses on concepts, principles, and techniques applicable to any technology environment and industry and establishes a baseline that can be enhanced by further formal training and additional real-world experience.

**UNIT - I 12 Periods**

**Introduction to BigData** :Introduction to BigData Platform - Traits of Big data - Challenges of Conventional Systems - Web Data - Evolution Of Analytic Scalability - Analytic Processes and Tools - Analysis Versus Reporting - Modern Data Analytic Tools - Statistical Concepts: Sampling Distributions - Re-Sampling - Statistical Inference - Prediction Error.

**UNIT - II 12 Periods**

**Data Analysis:** Regression Modelling - Multivariate Analysis - Bayesian Modelling - Inference and Bayesian Networks - Support Vector and Kernel Methods - Analysis of Time Series: Linear Systems Analysis - Nonlinear Dynamics - Rule Induction - Neural Networks: Learning And Generalization - Competitive Learning - Principal Component Analysis and Neural Networks - Fuzzy Logic: Extracting Fuzzy Models from Data - Fuzzy Decision Trees - Stochastic Search Methods,

**UNIT - III 12 Periods**

**Mining Data Streams:** Introduction To Streams Concepts - Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream - Filtering Streams - Counting Distinct Elements in a Stream - Estimating Moments - Counting Oneness in a Window - Decaying Window - Real time Analytics Platform(RTAP) Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.

**UNIT - IV 12 Periods**

**Frequent Itemsets and Clustering:** Mining Frequent Itemsets - Market Based Model - Apriori Algorithm - Handling Large Data Sets in Main Memory - Limited Pass Algorithm - Counting Frequent Itemsets in a Stream - Clustering Techniques - Hierarchical - K-Means - Clustering High Dimensional Data - CLIQUE And PROCLUS - Frequent Pattern based Clustering Methods - Clustering in Non-Euclidean Space - Clustering for Streams and Parallelism.

**UNIT - V 12 Periods**

**Hadoop and R for Visualization:** Background and fundamentals-moving data in and out of Hadoop-data serialization-applying MapReduce patterns to big data- streaming big data-integrating R and Hadoop for statistics and more-predictive analytics with Mahout- Hacking with Hive-Programming pipelines with pig - HBase-MySQL-NoSQL- RHadoop.

**REFERENCES**

|  |
| --- |
|  |

1. Michael Berthold, David J. Hand. *Intelligent Data Analysis.* Springer, 2007.
2. Anand Rajaraman, Jeffrey David Ullman. *Mining of Massive Datasets.* Cambridge University Press, 2012.
3. Bill Franks. *Taming the Big Data Tidal Wave: Finding OpportUNITies in Huge Data Streams with Advanced Analytics*. John Wiley & sons, First Edition, 2012.
4. Glenn J. Myatt. *Making Sense of Data.* John Wiley & Sons, 2007
5. Pete Warden. *Big Data Glossary.* O’Reilly, 2011.
6. Jiawei Han, Micheline Kamber. *Data Mining Concepts and Techniques.* Elsevier, Second Edition, 2006.
7. Alex Holmes. *Hadoop in Pracice.* Manning Publications, 2012.
8. A.Ohri. *R for Business Analytics.* Springer, 2012.
9. Prabhanjan Narayanachar Tattar. *R Statistical Application Development by Example Beginner's Guide*. Packt Publishing, 2013.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand the basic concepts in bigdata analytical processes and platform |
| Unit II | The learner will be able to understand the data analysis modeling |
| Unit III | The learner will be able to understand the data mining techniques for Big data analytics |
| Unit IV | The learner will be able to understand the concept of Frequent Itemsets and Clustering |
| Unit V | The learner will be able to use the tools such as R and R-Studio, Map-Reduce/Hadoop and select visualization techniques and tools to analyze big data and create statistical models |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E10**

**Semester:**

**INTERNET OF THINGS**

**Course Objectives:**

The learner familiarizes with functioning of devices, architecture, technologies and Application scenarios of Internet of things

**UNIT - I 12 Periods**

**Introduction to IoT**: Introduction - Definition & Characteristics of IoT - Physical Design of IoT - Things in IoT - IoT Protocols - Logical Design of IoT - IoT Functional blocks - IoT Communication Models - IoT Communication APIs - IoT Enabling Technologies - Wireless Sensor Networks - Cloud Computing - Big Data Analysis - Communication protocols - Embedded Systems - IoT Levels & Deployment Templates.

**IoT and M2M**: Introduction - M2M - Difference between IoT and M2M-SDN and NFV for IoT**.**

**Ubiquitous IoT Applications:** A Panoramic View of IoT Applications-Important Vertical IoT.

**UNIT - II 12 Periods**

**Four pillars of IoT**: The Horizontal, Verticals, and Four Pillars, M2M, RFID, WSN, SCADA. The DNA of IoT - Device, Connect and Manage - Device: Things That Talk - Connect: Via Pervasive Networks - Manage: To Create New Business Value.

**UNIT - III 12 Periods**

**Middleware and IoT**: An Overview of Middleware - Communication Middleware for IoT. **Protocol standardization of IoT** **:** Web of Things versus Internet of Things - IoT Protocol Standardization Efforts- Unified Data Standards: A Challenging Task.

**UNIT - IV 12 Periods**

**Architecture Standardization for WoT:** Platform Middleware for WoT - Unified Multitier WoT Architecture - WoT Portals and Business Intelligence - Challenges of IoT Information Security. **The Cloud of Things :** Cloud Middleware - NIST’s SPI Architecture and Cloud Standards - Cloud Providers and System.

**The Cloud of Things :** The Internet of Things and Cloud Computing, Mobile Cloud Computing. **MAI versus XaaS:** The Long Tail and the Big Switch - The Cloud of Things Architecture.

**UNIT - V 12 Periods**

**Thinking about Prototyping**: Prototypes and Production - Open Source versus Closed Source.

**Prototyping Embedded devices** **:** Electronics - Embedded Computing Basics - Arduino, Raspberry pi - Beagle Bone Black - Electric Imp - Other Notable Platforms.

**REFERENCES**

1. Arshdeep Bahga, Vijay Madisetti, f. *Internet of Things: A Hands on Approach.* Arshdeep Bagha & Vijay Madisetti, First Edition, 2014.
2. Honba Zhou. *The Internet of things in the Cloud: A Middleware Perspective*. CRC Press, First Edition, 2012.
3. Adrian McEwen, Hakim Cassimally. *Designing The Internet of things.* John Wiley and Sons, First Edition, 2014.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to discover the concepts of IoT Protocols and working principles of different types IoT enabling technologies |
| Unit II | The learner will be able to analyse difficulties in cloud of things architecture |
| Unit III | The learner will be able to understand the Protocol standardization of IoT and Middleware |
| Unit IV | The learner will be able to develop open source versus closed source in IoT |
| Unit V | The learner will be able to gain insight on Applications of IoT |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E11**

**Semester:**

**GAME THEORY**

**Course Objectives:**

To help the learner to apply game-theoretic analysis algorithm, both formally and intuitively, to negotiation and bargaining situations.

**UNIT - I 12 Periods**

**Strategic Form to Two Player Game:** The Cournot duopoly - Continuous improvement procedure -The Bertrand duopoly -The Hotelling duopoly -The Hotelling duopoly in 2D space -The Stackelberg duopoly -Convex games -Examples of bimatrix games -Randomization -2x2 Games- Games 2 × n and m × 2 -The Hotelling duopoly in 2D space with non-uniform distribution of buyers -Location problem in 2D space. **Zero Sum Games**: Introduction -Minimax and Maximin -Randomization -Games with discontinuous payoff functions -Convex-concave and linear-convex games -Convex games -Arbitration procedures -Two-point discrete arbitration procedures -Three-point discrete arbitration procedures with interval constraint -General discrete arbitration procedures.

**UNIT - II 12 Periods**

**Non-cooperative strategic-form n-player games :** Introduction-Convex games: The Cournot oligopoly -Polymatrix games -Potential games -Congestion games -Player-specific congestion games -Auctions -Wars of attrition -Duels, truels, and other shooting accuracy contests -Prediction games. **Extensive-form n-player games :** Introduction-Equilibrium in games with complete information -Indifferent equilibrium -Games with incomplete information -Total memory games.

**UNIT - III 12 Periods**

**Parlor games and sport games:** Introduction-Poker. A game-theoretic model -The poker model with variable bets -Preference. A game-theoretic model -The preference model with cards play -Twenty-one. A game-theoretic model -Soccer. A game-theoretic model of resource allocation. **Negotiation Models:** Introduction- Models of resource allocation -Negotiations of time and place of a meeting -Stochastic design in the cake cutting problem -Models of tournaments -Bargaining models with incomplete information -Reputation in negotiations.

**UNIT - IV 12 Periods**

**Optimal stopping games :** Introduction- Optimal stopping game: The case of two observations -Optimal stopping game: The case of independent observations -The game ΓN (G) under N ≥ 3 -Optimal stopping game with random walks -Best choice games -Best choice game with stopping before opponent -Best choice game with rank criterion. Lottery -Best choice game with rank criterion. Voting -Best mutual choice game.

**Cooperative Games**: Introduction-Equivalence of cooperative games -Imputations and core -Balanced games -The τ-value of a cooperative game -Nucleolus -The bankruptcy game -The Shapley vector -Voting games - Mutual influence of players.

**UNIT - V 12 Periods**

**Network Games:** Introduction-The KP-model of optimal routing - Pure strategy equilibrium: Braess’s paradox - Completely mixed equilibrium in the optimal routing problem - The price of anarchy -The Wardrop optimal routing model -The optimal routing model with parallel channels: Pigou model and Braess’s paradox -Potential in the optimal routing model with indivisible traffic -Social costs in the optimal routing model - The price of anarchy in the optimal routing model with divisible traffic -Potential in the Wardrop model with parallel channels -The price of anarchy in an arbitrary network

**Dynamic Games**: Introduction - Discrete-time dynamic games - Solution methods for optimal control problems with one player - maximum principle and Bellman equation in discrete and continuous time games.

**REFERENCES**

|  |
| --- |
|  |

1. Vladimir Mazalov. *Mathematical Game Theory and Applications*. John Wiley and Sons, First Edition, 2014.
2. Joel Watson. *Strategy-An Introduction to Game Theory.* W.W Norton & Company, Third Edition, 2013.

**Learning Outcomes**

|  |  |
| --- | --- |
| Unit I | The learner will be able to have knowledge about the two player game and zero sum games |
| Unit II | The learner will be able to have clear idea about non-cooperative and extensive form n players games |
| Unit III | The learner will be able to have an understanding of Parlor games, sport games and Negotiation Models |
| Unit IV | The learner will be able to have an understanding of Optimal stopping games and Cooperative Games |
| Unit V | The learner will be able to have an understanding of network games and dynamic games |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E12**

**Semester:**

**E12:** **ADVANCED COMPUTER ARCHITECTURE**

**Course Objectives:**

The objective of this course is to expose the learners to understand the advanced computer architectures that utilize parallelism via multiple processing units, understand the power and limitations of multiprocessor systems and gain knowledge and understanding of principles and practice in parallel computer architecture and computing, emphasizing both hardware and software challenges and the interactions between them.

**UNIT - I 12 Periods**

**Theory of Parallelism:** Parallel Computer Models: State of Computing - Multiprocessor and Multi-computers - Multi-vector and SIMD Computers - PRAM and VLSI Models - Architectural Development Tracks.

**Program and Network Properties:** Conditions of Parallelism - Program Partitioning and Scheduling - Program Flow Mechanisms - System Interconnect Architectures.

**UNIT - II 12 Periods**

**Hardware Technologies:** Processors and Memory Hierarchy: Advanced Processor Technology - Superscalar and Vector Processors - Memory Hierarchy Technology - Virtual Memory Technology.

**Bus, Cache, and Shared Memory:** Backplane Bus Systems - Cache Memory Organizations - Shared Memory Organizations - Sequential and Weak Consistency Models.

**UNIT - III 12 Periods**

**Pipelining and Superscalar Techniques:** Linear Pipeline Processors - Non-linear Pipeline Processors - Instruction Pipeline Processors - Arithmetic Pipeline Design - Superscalar and Super-pipeline Design.

**UNIT - IV 12 Periods**

**Instruction Level Parallelism:** Instruction-Level Parallelism: Concepts and Challenges - Basic Compiler techniques for exposing ILP - Reducing Branch costs and advanced branch prediction - Overcoming Data Hazards with Dynamic Scheduling - Dynamic Scheduling: Examples and the Algorithm - Hardware-Based Speculation - Exploiting ILP using Multiple Issue and static scheduling - Exploiting ILP using Dynamic Scheduling, Multiple Issue and Speculation, Advanced Techniques for Instruction Delivery and speculation - Studies of the Limitations of ILP.

**UNIT - V 12 Periods**

**Thread Level Parallelism:** Multithreading: Exploiting Thread-level Parallelism to improve Uni-Processor Throughput - The Intel Core i7 and ARM Cortex-A8 - Centralized Shared-Memory Architectures - Performance of Symmetric Shared-Memory Multiprocessors - Distributed Shared Memory and Directory based Coherence - Synchronization - Models of Memory Consistency – Multi-core Processors and their Performance.

**REFERENCES**

1. Kai Hwang, Naresh Jotwani. *Advanced Computer Architecture: Parallelism, Scalability, Programmability.* McGraw Hill International, Second Edition, 2011.
2. John L. Hennessy, David A. Patterson. *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann Title, Elsevier, Fifth Edition, 2011.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to gain a foundation for designing high-performance computers and for the development of support software and applications |
| Unit II | The learner will be able to understand concepts related to memory consistency models, cache coherence, interconnection networks, and latency tolerating techniques |
| Unit III | The learner will be able to explore the strategies for pipelining and superscalar design in processor development |
| Unit IV | The learner will be able to get exposure on the instruction level parallelism in high performance processors including superscalar execution, branch prediction, speculation, and dynamic scheduling |
| Unit V | The learner will be able to acquire knowledge on vector architectures and to compare the state of the art GPUs |

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **4** | **0** | **0** | **4** |

**Course Code: E13**

**Semester:**

**RANDOMIZED ALGORITHMS**

**Course Objectives:**

To understand the design of randomized algorithms, analyze the asymptotic performance of randomized algorithms, apply randomized algorithm design in common problems like Routing in a Parallel Computer, Linear Programming, Byzantine Agreement etc.

**UNIT - I 12 Periods**

**Introduction:** Min-Cut algorithm-Las Vegas and Monte Carlo-Binary Planar Partitions-Probabilistic Recurrence-Computation Model and Complexity Classes. Game Theoretic Techniques: Game Tree Evaluation-Minimax Principle-Randomness and Non Uniformity. Moments and Deviations: Occupancy Problems-Markov and Chebyshev Inequalities-Randomized Selection-Two Point Sampling-Stable Marriage Problem-Coupon Collectors Problem.

**UNIT - II 12 Periods**

**Tail Inequalities:** Chernoff Bound-Routing in a parallel Computer-Wiring Problem-Martingales. Probabilistic Method: Overview-Maximum Satisfiability-Expanding Graphs-Routing-Lovasz Local Lemma-Method of Conditional Probabilities. Markov Chains and Random Walks: 2 SAT Example-Markov Chains-Random walks on Graphs-Electrical Networks-Cover times-Graph Connectivity-Expanders and Rapidly mixing Random walks-Probability amplification by Random Walks on Expanders.

**UNIT - III 12 Periods**

**Algebraic Techniques:** Fingerprinting and Freivald's Technique-Verifying Polynomial Identities-Perfect Matchings in Graphs-Verifying Equality of Strings-Comparison of Fingerprinting Techniques-Pattern Matching-Interactive Proof Systems-PCP and Efficient Proof Verification. Data Structures: Fundamental Data Structuring Problem-Random Treaps-Skip lists-Hash tables-Hashing with O(1) Search Time. Geometric Algorithms and Linear Programming: Randomized Incremental Construction-Convex Hulls in the Plane-Duality-Half space Intersections-Delaunay Triangulations-Trapezoidal decompositions-Binary space partitions-Diameter of a Point Set-Random Sampling-Linear Programming.

**UNIT - IV 12 Periods**

**Graph Algorithms:** All Pairs Shortest Paths-Min-Cut Problem-Minimum Spanning Trees. Approximate Counting: Randomized Approximation Schemes-DNF Counting Problem-Approximating the Permanent-Volume Estimation.

**UNIT - V 12 Periods**

**Parallel and Distributed Algorithms:** PRAM Model-Sorting on a PRAM-Maximal Independent Sets-Perfect Matchings-Choice Coordination Problem- Byzantine Agreement. Online Algorithms: Online Paging Problem-Adversary Models-Paging against an Oblivious Adversary-Relating the Adversaries-Adaptive Online Adversary-k-Server Problem.

**REFERENCES**

1. Rajeev Motwani, Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 2006.
2. Devdatt P. Dubhashi , Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms.* Cambridge University Press, First edition, 2009.

**LEARNING OUTCOMES**

|  |  |
| --- | --- |
| Unit I | The learner will be able to understand about LasVegas and MonteCarlo methods. They will apply these methods respectively to solve Random Quick Sort and MinCut Problems |
| Unit II | The learner will be able to understand tail inequalities, probabilistic methods and random walks. They will apply these concepts to solve wiring, routing in a parallel computer and resistive electrical network problems |
| Unit III | The learner will be able to understand the various algebraic techniques and apply the same to determine perfect matching and finger print verification. The learner will apply the randomized algorithmic techniques to create Skip lists, Hash tables, Delaunay Triangulations and Linear Programming |
| Unit IV | The learner will be able to apply randomized algorithmic techniques to solve all pairs shortest paths, minimum spanning trees, DNF counting and volume estimation problems |
| Unit V | The learner will be able to understand the design of randomized parallel, distributed and online algorithms. The Learners will apply these design techniques to sorting, Maximal Independent Sets, Perfect Matching, Choice Coordination, Byzantine Agreement, Paging and k-server problems |