

## Cryptography and Network Security

### Course objectives:

The main objective of this course is to teach students to understand and how to address various software security problems in a secure and controlled environment. During this course the students will gain knowledge (both theoretical and practical) in various kinds of software security problems, and techniques that could be used to protect the software from security threats. The students will also learn to understand the “modus operandi” of adversaries; which could be used for increasing software dependability.

### Course outcomes:

1. be able to individually reason about software security problems and protection techniques on both an abstract and a more technically advanced level.
2. be able to individually explain how software exploitation techniques, used by adversaries, function and how to protect against them.

### Syllabus:

#### UNIT I : Classical Encryption Techniques

**Objectives:** *The Objectives of this unit is to present an overview of the main concepts of cryptography, understand the threats & attacks, understand ethical hacking.*

**Introduction:** Security attacks, services & mechanisms, Symmetric Cipher Model, Substitution Techniques, Transportation Techniques, Cyber threats and their defense( Phishing Defensive measures, web based attacks, SQL injection & Defense techniques)(TEXT BOOK 2), Buffer overflow & format string vulnerabilities, TCP session hijacking(ARP attacks, route table modification) UDP hijacking ( man-in-the-middle attacks)(TEXT BOOK 3).

#### UNIT II: Block Ciphers & Symmetric Key Cryptography

**Objectives:** *The Objectives of this unit is to understand the difference between stream ciphers & block ciphers, present an overview of the Feistel Cipher and explain the encryption and decryption, present an overview of DES, Triple DES, Blowfish, IDEA.*

Traditional Block Cipher Structure, DES, Block Cipher Design Principles, AES-Structure, Transformation functions, Key Expansion, Blowfish, CAST-128, IDEA, Block Cipher Modes of Operations

#### UNIT III: Number Theory & Asymmetric Key Cryptography

**Objectives:** *Presents the basic principles of public key cryptography, Distinct uses of public key cryptosystems*

**Number Theory:** Prime and Relatively Prime Numbers, Modular Arithmetic, Fermat’s and Euler’s Theorems, The Chinese Remainder theorem, Discrete logarithms.

**Public Key Cryptography:** Principles, public key cryptography algorithms, RSA Algorithms, Diffie Hellman Key Exchange, Elgamal encryption & decryption, Elliptic Curve Cryptography.

#### UNIT IV : Cryptographic Hash Functions & Digital Signatures

**Objectives:** *Present overview of the basic structure of cryptographic functions, Message Authentication Codes, Understand the operation of SHA-512, HMAC, Digital Signature*

Application of Cryptographic hash Functions, Requirements & Security, Secure Hash Algorithm, Message Authentication Functions, Requirements & Security, HMAC & CMAC. Digital Signatures, NIST Digital Signature Algorithm. Key management & distribution.

## **UNIT V: User Authentication, Transport Layer Security & Email Security**

**Objectives:** *Present an overview of techniques for remote user authentication, Kerberos, Summarize Web Security threats and Web traffic security approaches, overview of SSL & TLS. Present an overview of electronic mail security.*

**User Authentication:** Remote user authentication principles, Kerberos

**Transport Level Security:** Web Security Requirements, Secure Socket Layer (SSL) and Transport Layer Security (TLS), Secure Shell (SSH)

**Electronic Mail Security:** Pretty Good Privacy (PGP) and S/MIME.

## **UNIT VI: IP Security & Intrusion Detection Systems**

**Objectives:** *Provide an overview of IP Security, concept of security association, Intrusion Detection Techniques*

**IP Security:** IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management.

**Intrusion detection:** Overview, Approaches for IDS/IPS, Signature based IDS, Host based IDS/IPS. (TEXT BOOK 2)

### **TEXT BOOKS:**

1. Cryptography & Network Security: Principles and Practices, William Stallings, PEA, Sixth edition.
2. Introduction to Computer Networks & Cyber Security, Chwan Hwa Wu, J. David Irwin, CRC press
3. Hack Proofing your Network, Russell, Kaminsky, Forest Puppy, Wiley Dreamtech.

### **REFERENCE BOOKS:**

1. Everyday Cryptography, Fundamental Principles & Applications, Keith Martin, Oxford
2. Network Security & Cryptography, Bernard Menezes, Cengage, 2010

## UML & Design Patterns

### Course Objectives:

The focus of this course is on design rather than implementation.

1. Introducing the Unified Process and showing how UML can be used within the process.
2. Presenting a comparison of the major UML tools for industrial-strength development.
3. introduction to design patterns, practical experience with a selection of central patterns.

### Course Outcomes:

Students successfully completing this course will be able to:

1. identify the purpose and methods of use of common object-oriented design patterns
2. select and apply these patterns in their own designs for simple programs
3. represent the data dependencies of a simple program using UML
4. represent user and programmatic interactions using UML
5. create design documentation outlining the testable and complete design of a simple program
6. produce and present documents for the purpose of capturing software requirements and specification
7. produce plans to limit risks specific to software designed for use in a particular social context

### Syllabus:

**Unit I: Introduction :** Introduction to OOAD; typical activities / workflows / disciplines in OOAD, Introduction to iterative development and the Unified Process, Introduction to UML; mapping disciplines to UML artifacts, Introduction to Design Patterns - goals of a good design, Introducing a case study & MVC architecture

**Unit II: Inception:** Artifacts in inception, Understanding requirements - the FURPS model, Understanding Use case model - introduction, use case types and formats, Writing use cases - goals and scope of a use case, elements / sections of a use case, Use case diagrams, Use cases in the UP context and UP artifacts, Identifying additional requirements, Writing requirements for the case study in the use case model

**Unit III: Elaboration:** System sequence diagrams for use case model, Domain model : identifying concepts, adding associations, adding attributes, Interaction Diagrams, Introduction to GRASP design Patterns ,Design Model: Use case realizations with GRASP patterns, Design Class diagrams in each MVC layer  
Mapping Design to Code, Design class diagrams for case study and skeleton code

**Unit 4: More Design Patterns:** Fabrication, Indirection, Singleton, Factory, Facade, Publish-Subscribe

**Unit 5: More UML diagrams :** State-Chart diagrams, Activity diagrams, Component Diagrams, Deployment diagrams, Object diagrams

**Unit 6: Advanced concepts in OOAD :** Use case relationships, Generalizations  
Domain Model refinements, Architecture, Packaging model elements

### Textbooks:

1. 'Applying UML and patterns' by Craig Larman, Pearson
2. Object-Oriented Analysis & Design with the Unified Process by Satzinger, Jackson & Burd Cengage Learning
3. 'UML distilled' by Martin Fowler , Addison Wesley, 2003

### Reference:

1. O'reilly 's 'Head-First Design Patterns' by Eric Freeman et al, Oreilly
2. UML 2 Toolkit, by Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: WILEY'-Dreamtech India Pvt. Lid.

## Mobile Computing

### Course Objective:

- 1) To make the student understand the concept of mobile computing paradigm, its novel applications and limitations.
- 2) To understand the typical mobile networking infrastructure through a popular GSM protocol
- 3) To understand the issues and solutions of various layers of mobile networks, namely MAC layer, Network Layer & Transport Layer
- 4) To understand the database issues in mobile environments & data delivery models.
- 5) To understand the ad hoc networks and related concepts.
- 6) To understand the platforms and protocols used in mobile environment.

### Course Outcomes:

- 1) Able to think and develop new mobile application.
- 2) Able to take any new technical issue related to this new paradigm and come up with a solution(s).
- 3) Able to develop new ad hoc network applications and/or algorithms/protocols.
- 4) Able to understand & develop any existing or new protocol related to mobile environment

### Syllabus:

#### UNIT I

**Introduction:** Mobile Communications, Mobile Computing – Paradigm, Promises/Novel Applications and Impediments and Architecture; Mobile and Handheld Devices, Limitations of Mobile and Handheld Devices.

GSM – Services, System Architecture, Radio Interfaces, Protocols, Localization, Calling, Handover, Security, New Data Services, GPRS.

#### UNIT –II

**(Wireless) Medium Access Control (MAC) :** Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA, Wireless LAN/(IEEE 802.11)

#### UNIT –III

**Mobile Network Layer :** IP and Mobile IP Network Layers, Packet Delivery and Handover Management, Location Management, Registration, Tunneling and Encapsulation, Route Optimization, DHCP.

#### UNIT –IV

**Mobile Transport Layer :** Conventional TCP/IP Protocols, Indirect TCP, Snooping TCP, Mobile TCP, Other Transport Layer Protocols for Mobile Networks.

**Database Issues :** Database Hoarding & Caching Techniques, Client-Server Computing & Adaptation, Transactional Models, Query processing, Data Recovery Process & QoS Issues.

#### UNIT V

**Data Dissemination and Synchronization :** Communications Asymmetry, Classification of Data Delivery Mechanisms, Data Dissemination, Broadcast Models, Selective Tuning and Indexing Methods, Data Synchronization – Introduction, Software, and Protocols.

#### UNIT VI

**Mobile Ad hoc Networks (MANETs) :** Introduction, Applications & Challenges of a MANET, Routing, Classification of Routing Algorithms, Algorithms such as DSR, AODV, DSDV, etc. , Mobile Agents, Service Discovery.

**Protocols and Platforms for Mobile Computing :** WAP, Bluetooth, XML, J2ME, JavaCard, PalmOS, Windows CE, SymbianOS, Linux for Mobile Devices,Android.

**Text Books:**

1. Jochen Schiller, “Mobile Communications”, Addison-Wesley, Second Edition, 2009.
2. Raj Kamal, “Mobile Computing”, Oxford University Press, 2007, ISBN: 0195686772

**Reference Book:**

1. ASOKE K TALUKDER, HASAN AHMED, ROOPA R YAVAGAL, “Mobile Computing, Technology Applications and Service Creation” Second Edition, Mc Graw Hill.
2. UWE Hansmann, Lothar Merk, Martin S. Nocklous, Thomas Stober, “Principles of Mobile Computing,” Second Edition, Springer.

IV Year – I SEMESTER

T	P	C
3+1	0	3

**Elective - I**

**Software Testing Methodologies**

**Course Objectives:**

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
5. To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
6. To understand software test automation problems and solutions.
7. To learn how to write software testing documents, and communicate with engineers in various forms.
8. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

**Course Outcomes:**

By the end of the course, the student should:

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems
7. Have an ability to use software testing methods and modern software testing tools for their testing projects.

**Syllabus:**

**UNIT I:**

**Software Testing:** Introduction, Evolution, Myths & Facts, Goals, Psychology, Definition, Model for testing, Effective Vs Exhaustive Software Testing.

**Software Testing Terminology and Methodology:** Software Testing Terminology, Software Testing Life Cycle, relating test life cycle to development life cycle Software Testing Methodology.

#### **UNIT II:**

**Verification and Validation:** Verification & Validation Activities, Verification, Verification of Requirements, High level and low level designs, How to verify code, Validation

**Dynamic Testing I: Black Box testing techniques:** Boundary Value Analysis, Equivalence class Testing, State Table based testing, Decision table based testing, Cause-Effect Graphing based testing, Error guessing

#### **UNIT III:**

**Dynamic Testing II: White-Box Testing:** need, Logic coverage criteria, Basis path testing, Graph matrices, Loop testing, data flow testing, mutation testing

**Static Testing:** inspections, Structured Walkthroughs, Technical reviews

#### **UNIT IV:**

**Validation activities:** Unit testing, Integration Testing, Function testing, system testing, acceptance testing

**Regression testing:** Progressives Vs regressive testing, Regression testability, Objectives of regression testing, When regression testing done?, Regression testing types, Regression testing techniques

#### **UNIT V:**

**Efficient Test Suite Management:** Test case design Why does a test suite grow, Minimizing the test suite and its benefits, test suite prioritization, Types of test case prioritization, prioritization techniques, measuring the effectiveness of a prioritized test suite

**Software Quality Management:** Software Quality metrics, SQA models

**Debugging:** process, techniques, correcting bugs, Basics of testing management tools, test link and Jira

#### **UNIT VI:**

**Automation and Testing Tools:** need for automation, categorization of testing tools, selection of testing tools, Cost incurred, Guidelines for automated testing, overview of some commercial testing tools.

**Testing Object Oriented Software:** basics, Object oriented testing

**Testing Web based Systems:** Challenges in testing for web based software, quality aspects, web engineering, testing of web based systems, Testing mobile systems

#### **Text Books:**

1. Software Testing, Principles and Practices, Naresh Chauhan, Oxford
2. Foundations of Software testing, Aditya P Mathur, 2ed, Pearson
3. Software Testing- Yogesh Singh, CAMBRIDGE

#### **Reference books:**

1. *Software testing techniques - Baris Beizer, International Thomson computer press, second edition.*
2. Software Testing, Principles, techniques and Tools, M G Limaye, TMH
3. Effective Methods for Software testing, Willian E Perry, 3ed, Wiley

# Simulation Modeling

## Course Objectives: □

1. Introduce computer simulation technologies and techniques, provides the foundations for the student to understand computer simulation needs, and to implement and test a variety of simulation and data analysis libraries and programs. This course focusses what is needed to build simulation software environments, and not just building simulations using preexisting packages.
2. Introduce concepts of modeling layers of society's critical infrastructure networks.
3. Build tools to view and control simulations and their results.

## Course Outcomes:

1. provide a strong foundation on concept of simulation, and modeling.
2. understand the techniques of random number generations.
3. understand the techniques of testing randomness.
4. design simulation models for various case studies like inventory, traffic flow networks, etc.
5. practice on simulation tools and impart knowledge on building simulation systems.

## Syllabus:

### UNIT-I:

System models: Concepts, continuous and discrete systems, System modeling, types of models, subsystems, system study.

### UNIT-II:

System Simulation: Techniques, comparison of simulation and analytical methods, types of simulation, Distributed log models, cobweb models.

### UNIT-III:

Continuous system Simulation: Numerical solution of differential equations, Analog Computers, Hybrid Computers, continuous system simulation languages CSMP, system dynamic growth models, logistic curves.

### UNIT-IV:

Probability concepts in simulation: Monte Carlo techniques, stochastic variables, probability functions, Random Number generation algorithms.

### UNIT-V:

Queuing Theory: Arrival pattern distributions, servicing times, queuing disciplines, measure of queues, mathematical solutions to queuing problems.

Discrete System Simulation: Events, generation of arrival patterns, simulation programming tasks, analysis of simulation output.

### UNIT-VI:

GPSS & SIMSCRIPT: general description of GPSS and SIMSCRIPT, programming in GPSS & SIMSCRIPT, Data structures, Implementation of activities, events and queues, Event scanning, simulation algorithms in GPSS and SIMSCRIPT.

□

## TEXT BOOKS

1. Geoffrey Gordon, "System Simulation", 2nd Edition, Prentice Hall, India, 2002.
2. Narsingh Deo, "System Simulation with Digital Computer", Prentice Hall, India, 2001.

## REFERENCES

1. Jerry Banks and John S. Carson, Barry L. Nelson, David M. Nicol, "Discrete Event System Simulation", 3rd Edition, Prentice Hall, India, 2002.

2. Shannon, R.E. Systems simulation, The art and science, Prentice Hall, 1975.
3. Thomas J. Schriber, Simulation using GPSS, John Wiley, 1991



# Information Retrieval Systems

## COURSE OBJECTIVES

- To provide the foundation knowledge in information retrieval.
- To equip students with sound skills to solve computational search problems.
- To appreciate how to evaluate search engines.
- To appreciate the different applications of information retrieval techniques in the Internet or Web environment.
- To provide hands-on experience in building search engines and/or hands-on experience in evaluating search engines.

## COURSE OUTCOMES

After completing the course student will be able to:

- Identify basic theories in information retrieval systems
- Identify the analysis tools as they apply to information retrieval systems
- Understands the problems solved in current IR systems
- Describes the advantages of current IR systems
- Understand the difficulty of representing and retrieving documents.
- Understand the latest technologies for linking, describing and searching the web.
- Explain the concepts of indexing, vocabulary, normalization and dictionary in information retrieval.
- Evaluate information retrieval algorithms, and give an account of the difficulties of evaluation
- Use different information retrieval techniques in various application areas
- Apply IR principles to locate relevant information large collections of data
- Analyze performance of retrieval systems when dealing with unmanaged data sources
- Implement retrieval systems for web search tasks.
- Understand and apply the basic concepts of information retrieval;
- Appreciate the limitations of different information retrieval techniques;
- Write programs to implement search engines;
- Evaluate search engines;
- Develop skills in problem solving using systematic approaches;
- Solve complex problems in groups and develop group work.

## SYLLABUS:

### Unit I:

**Introduction to Information Storage and Retrieval System:** Introduction, Domain Analysis of IR systems and other types of Information Systems, IR System Evaluation.

Introduction to Data Structures and Algorithms related to Information Retrieval □ Basic Concepts, Data structures, Algorithms

□

### Unit II:

**Inverted files:** Introduction, Structures used in Inverted Files, Building Inverted file using a sorted array, Modifications to Basic Techniques.

□

### Unit III:

**Signature Files:** Introduction, Concepts of Signature Files, Compression, Vertical Partitioning, Horizontal Partitioning.

□

#### **Unit IV:**

**New Indices for Text:** PAT Trees and PAT Arrays: Introduction, PAT Tree structure, algorithms on the PAT Trees, Building PAT trees as PATRICA Trees, PAT representation as arrays.

□

#### **Unit V:**

**Stemming Algorithms:** Introduction, Types of Stemming Algorithms, Experimental Evaluations of Stemming to Compress Inverted Files

□

#### **Unit VI:**

**Thesaurus Construction:** Introduction, Features of Thesauri, Thesaurus Construction, Thesaurus construction from Texts, Merging existing Thesauri

□

#### **TEXT BOOK :**

1. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
2. Modern Information Retrieval By Yates Pearson Education.
3. Information Storage & Retrieval By Robert Korfhage – John Wiley & Sons.

□

#### **REFERENCES :**

1. Kowalski, Gerald, Mark T Maybury: Information Retrieval Systems: Theory and Implementation, Kluwer Academic Press, 1997.
2. Information retrieval Algorithms and Heuristics, 2ed, Springer

# Artificial Intelligence

## Course Objectives:

1. To have a basic proficiency in a traditional AI language including an ability to write simple to intermediate programs and an ability to understand code written in that language.
2. To have an understanding of the basic issues of knowledge representation and blind and heuristic search, as well as an understanding of other topics such as minimax, resolution, etc. that play an important role in AI programs.
3. To have a basic understanding of some of the more advanced topics of AI such as learning, natural language processing, agents and robotics, expert systems, and planning

## Course Outcomes:

After completing this course, students should be able to:

1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
2. Formalize a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, as a Markov decision process, etc).
3. Implement basic AI algorithms (e.g., standard search algorithms or dynamic programming).
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.

## Syllabus:

### UNIT-I:

**Introduction to artificial intelligence:** Introduction ,history, intelligent systems, foundations of AI, applications, tic-tac-tie game playing, development of ai languages, current trends in AI

### UNIT-II:

**Problem solving: state-space search and control strategies :** Introduction, general problem solving, characteristics of problem, exhaustive searches, heuristic search techniques, iterative-deepening a\*, constraint satisfaction

**Problem reduction and game playing:** Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games

### UNIT-III:

**Logic concepts:** Introduction, propositional calculus, propositional logic, natural deduction system, axiomatic system, semantic tableau system in propositional logic, resolution refutation in propositional logic, predicate logic

### UNIT-IV:

**Knowledge representation:** Introduction, approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, knowledge representation using frames **advanced knowledge representation techniques:** Introduction, conceptual dependency theory, script structure, cyc theory, case grammars, semantic web

### UNIT-V:

**Expert system and applications:** Introduction phases in building expert systems, expert system versus traditional systems, rule-based expert systems blackboard systems truth maintenance systems, application of expert systems, list of shells and tools

### UNIT-VI:

**Uncertainty measure: probability theory:** Introduction, probability theory, Bayesian belief networks, certainty factor theory, dempster-shafer theory

**Fuzzy sets and fuzzy logic:** Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.

**TEXT BOOKS:**

1. Artificial Intelligence- Saroj Kaushik, CENGAGE Learning,
2. Artificial intelligence, A modern Approach , 2<sup>nd</sup> ed, Stuart Russel, Peter Norvig, PEA
3. Artificial Intelligence- Rich, Kevin Knight, Shiv Shankar B Nair, 3<sup>rd</sup> ed, TMH
4. Introduction to Artificial Intelligence, Patterson, PHI

**REFERNCE BOOKS:**

1. Atificial intelligence, structures and Strategies for Complex problem solving, -George F Lugar, 5<sup>th</sup> ed, PEA
2. Introduction to Artificial Intelligence, Ertel, Wolf Gang, Springer
3. Artificial Intelligence, A new Synthesis, Nils J Nilsson, Elsevier

# Multimedia Computing

## Course objectives:

To provide the foundation knowledge of multimedia computing, e.g. media characteristics, compression standards, multimedia representation, data formats, multimedia technology development.

## Course outcomes:

1. understand the characteristics of different media; understand the representations of different multimedia data; understand different data formats; be able to take into considerations in multimedia system designs;
2. understand the characteristics of human's visual system; understand the characteristics of human's audio system; be able to take into considerations in multimedia techniques design and implementation;
3. understand different compression principles; understand different compression techniques; understand different multimedia compression standards; be able to design and develop multimedia systems according to the requirements of multimedia applications.
4. program multimedia data and be able to design and implement media applications; □

## Syllabus:

### UNIT-I:

Fundamental concepts in Text and Image: Multimedia and hypermedia, World Wide Web, overview of multimedia software tools. Graphics and image data representation graphics/image data types, file formats, Color in image and video: color science, color models in images, color models in video.

### UNIT-II:

Fundamental concepts in video and digital audio: Types of video signals, analog video, digital video, digitization of sound, MIDI, quantization and transmission of audio.

### UNIT-III:

**Multimedia data compression I:** Lossless compression algorithm: Run-Length Coding, Variable Length Coding, Dictionary Based Coding, Arithmetic Coding, Lossless Image Compression,

### UNIT-IV:

**Multimedia data compression II:** Lossy compression algorithm: Quantization, Transform Coding, Wavelet-Based Coding, Embedded Zerotree of Wavelet Coefficients Set Partitioning in Hierarchical Trees (SPIHT).

### UNIT-V:

**Basic Video Compression Techniques:** Introduction to video compression, video compression based on motion compensation, search for motion vectors, MPEG, Basic Audio Compression Techniques.

### UNIT-VI:

Multimedia Networks: Basics of Multimedia Networks, Multimedia Network Communications and Applications: Quality of Multimedia Data Transmission, Multimedia over IP, Multimedia over ATM Networks, Transport of MPEG-4, Media-on-Demand (MOD).

## TEXT BOOKS:

1. Fundamentals of Multimedia by Ze-Nian Li and Mark S. Drew Pearson Education.

## REFERENCE BOOKS:

1. Digital Multimedia, Nigel Chapman and Jenny Chapman, Wiley-Dreamtech
2. Macromedia Flash MX Professional 2004 Unleashed, Pearson.
3. Multimedia and communications Technology, Steve Heath, Elsevier (Focal Press).
4. Multimedia Applications, Steinmetz, Nahrstedt, Springer.
5. Multimedia Basics by Weixel Thomson
6. Multimedia Technology and Applications, David Hilman, Galgotia

# High Performance Computing

## Course Objectives:

This course covers the design of advanced modern computing systems. In particular, the design of modern microprocessors, characteristics of the memory hierarchy, and issues involved in multi-threading and multi-processing are discussed. The main objective of this course is to provide students with an understanding and appreciation of the fundamental issues and tradeoffs involved in the design and evaluation of modern computers

## Course Outcomes:

1. Understand the concepts and terminology of high performance computing.
2. Can write and analyze the behavior of high performance parallel programs for distributed memory architectures (using MPI).
3. Can write and analyze the behavior of high performance parallel programs for shared memory architectures (using Pthreads and OpenMP).
4. Can write simple programs for the GPU.
5. Can independently study, learn about, and present some aspect of high performance computing.

## Syllabus:

### UNIT I:

Introduction to Parallel hardware and software, need for high performance systems and Parallel Programming, SISD, SIMD, MISD, MIMD models, Performance issues.

### UNIT II:

Processors, PThreads, Thread Creation, Passing arguments to Thread function, Simple matrix multiplication using Pthreads, critical sections, mutexes, semaphores, barriers and conditional variables, locks, thread safety, simple programming assignments.

### UNIT III:

OpenMP Programming: introduction, reduction clause, parallel for-loop scheduling, atomic directive, critical sections and locks, private directive, Programming assignments, n body solvers using openMP.

### UNIT IV:

Introduction to MPI programming: MPI primitives such as MPI\_Send, MPI\_Recv, MPI\_Init, MPI\_Finalize, etc., Application of MPI to Trapezoidal rule, Collective Communication primitives in MPI, MPI derived datatypes, Performance evaluation of MPI programs, Parallel sorting algorithms, Tree search solved using MPI, Programming Assignments.

### UNIT V:

Introduction to GPU computing, Graphics pipelines, GPGPU, Data Parallelism and CUDA C Programming, CUDA Threads Organization, Simple Matrix multiplication using CUDA, CUDA memories.

### UNIT VI:

Bench Marking and Tools for High Performance Computing Environments, Numerical Linear Algebra Routines BLAS for Parallel Systems evaluation.

## Text Books:

1. An Introduction to Parallel Programming, Peter S Pacheco, Elsevier, 2011
2. Programming Massively Parallel Processors, Kirk & Hwu, Elsevier, 2012

**Reference Books:**

1. CUDA by example: An introduction to General Purpose GPU Programming, Jason, Sanders, Edward Kandrit, Perason, 2011
2. CUDA Programming, Shame Cook, Elsevier
3. High Performance Heterogeneous Computing, Jack Dongarra, Alexey & Lastovetsky , Wiley
4. Parallel computing theory and practice, Michel J.Quinn, TMH

## Elective - II

### Digital Forensics

**Course Objectives:**

This course is intended to provide students with greater depth of study in a number of key topics in the area of computer security in society: cybercrime, computer and forensics, analysis

**Course Outcomes:**

1. Understand financial and accounting forensics, and explain their role in preventing various forms of fraud.
2. Distinguish various types of computer crime, and use computer forensic techniques to identify the digital fingerprints associated with criminal activities

**Syllabus:****Unit-I:**

**Computer Forensics and Investigations:** Understanding Computer Forensics, Preparing for Computer Investigations, Taking A Systematic Approach, Procedure for Corporate High-Tech Investigations, Understanding Data Recovery Workstations and Software,

**Investor's Office and Laboratory:** Understanding Forensics Lab Certification Requirements, Determining the Physical Requirements for a Computer Forensics Lab, Selecting a Basic Forensic Workstation

**Unit-II:**

**Data Acquisition:** Understanding Storage Formats for Digital Evidence, Determining the Best Acquisition Method, Contingency Planning for Image Acquisitions, Using Acquisition Tools, Validating Data Acquisition, Performing RAID Data Acquisition, Using Remote Network Acquisition Tools, Using Other Forensics Acquisition Tools

**Unit-III:**

**Processing Crime and Incident Scenes:** Identifying Digital Evidence, Collecting the Evidence in Private-Sector Incident Scenes, Processing law Enforcement Crime Scenes, Preparing for a Search, Securing a Computer Incident or Crime Scene, Sizing Digital evidence at the Scene, Storing Digital evidence, obtaining a Digital Hash.

**Unit-IV:**

**Current Computer Forensics Tools:** Evaluating Computer Forensics Tool Needs, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software

**Computer Forensics Analysis and Validation:** Determining What Data to Collect and Analyze, Validating Forensic Data, Addressing Data-Hiding Techniques, Performing Remote Acquisition

**Unit-V:**

**Recovering Graphics and Network Forensics:** Recognizing a Graphics File, Understanding Data Compression, Locating and Recovering Graphics Files, Understanding Copyright Issues with Graphics, Network Forensic, Developing Standard Procedure for Network Forensics, Using Network Tools, Examining Hiney Project

**Unit-VI:**

**E-mail Investigations Cell Phone and Mobile Device Forensics:** Exploring the Role of E-mail in Investigations, Exploring the Role of Client and Server in E-mail, Investigating E-mail Crimes and Violations, Understanding E-mail Servers, Using Specialized E-mail Forensics Tools, Understanding Mobile Device Forensics, Understanding Acquisition Procedure for Cell Phones and Mobile Devices

**TEXT BOOK:**

1. Nelson, Phillips Enfinger,Stewart, “ Computer Forensics and Investigations, Cengage Learning



# Hadoop and Big Data

## Course Objectives:

- Optimize business decisions and create competitive advantage with Big Data analytics
- Introducing Java concepts required for developing map reduce programs
- Derive business benefit from unstructured data
- Imparting the architectural concepts of Hadoop and introducing map reduce paradigm
- To introduce programming tools PIG & HIVE in Hadoop ecosystem.

## Course Outcomes:

- Preparing for data summarization, query, and analysis.
- Applying data modelling techniques to large data sets
- Creating applications for Big Data analytics
- Building a complete business data analytic solution

## Unit 1:

Data structures in Java: Linked List, Stacks, Queues, Sets, Maps; Generics: Generic classes and Type parameters, Implementing Generic Types, Generic Methods, Wrapper Classes, Concept of Serialization

Reference:

Big Java 4th Edition, Cay Horstmann, Wiley John Wiley & Sons, INC

## Unit 2:

Working with Big Data: Google File System, Hadoop Distributed File System (HDFS) – Building blocks of Hadoop (Namenode, Datanode, Secondary Namenode, JobTracker, TaskTracker), Introducing and Configuring Hadoop cluster (Local, Pseudo-distributed mode, Fully Distributed mode), Configuring XML files.

References:

Hadoop: The Definitive Guide by Tom White, 3<sup>rd</sup> Edition, O'reilly

Hadoop in Action by Chuck Lam, MANNING Publ.

## Unit 3:

Writing MapReduce Programs: A Weather Dataset, Understanding Hadoop API for MapReduce Framework (Old and New), Basic programs of Hadoop MapReduce: Driver code, Mapper code, Reducer code, RecordReader, Combiner, Partitioner

Reference:

Hadoop: The Definitive Guide by Tom White, 3<sup>rd</sup> Edition, O'reilly

## Unit 4:

Hadoop I/O: The Writable Interface, WritableComparable and comparators, Writable Classes: Writable wrappers for Java primitives, Text, BytesWritable, NullWritable, ObjectWritable and GenericWritable, Writable collections, Implementing a Custom Writable: Implementing a RawComparator for speed, Custom comparators

Reference:

Hadoop: The Definitive Guide by Tom White, 3<sup>rd</sup> Edition, O'reilly

## Unit 5:

Pig: Hadoop Programming Made Easier

Admiring the Pig Architecture, Going with the Pig Latin Application Flow, Working through the ABCs of Pig Latin, Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin

Reference:

Hadoop for Dummies by Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk,Bruce Brown, Rafael Coss

### **Unit 6:**

Applying Structure to Hadoop Data with Hive:

Saying Hello to Hive, Seeing How the Hive is Put Together, Getting Started with Apache Hive, Examining the Hive Clients, Working with Hive Data Types, Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works, Querying and Analyzing Data

References:

Hadoop for Dummies by Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk,Bruce Brown, Rafael Coss

### **Text Books:**

1. Big Java 4th Edition, Cay Horstmann, Wiley John Wiley & Sons, INC
2. Hadoop: The Definitive Guide by Tom White, 3<sup>rd</sup> Edition, O'reilly
3. Hadoop in Action by Chuck Lam, MANNING Publ.
4. Hadoop for Dummies by Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk,Bruce Brown, Rafael Coss

### **References:**

1. Hadoop in Practice by Alex Holmes, MANNING Publ.
2. Hadoop MapReduce Cookbook,Srinath Perera, Thilina Gunarathne

### **Software Links:**

1. Hadoop:<http://hadoop.apache.org/>
  2. Hive:<https://cwiki.apache.org/confluence/display/Hive/Home>
- Piglatin:<http://pig.apache.org/docs/r0.7.0/tutorial.html>

# Software Project Management

## Course Objectives:

1. To study how to plan and manage projects at each stage of the software development life cycle (SDLC)
2. To train software project managers and other individuals involved in software project planning and tracking and oversight in the implementation of the software project management process.
3. To understand successful software projects that support organization's strategic goals

## Course Outcomes:

1. To match organizational needs to the most effective software development model
2. To understand the basic concepts and issues of software project management
3. To effectively Planning the software projects
4. To implement the project plans through managing people, communications and change
5. To select and employ mechanisms for tracking the software projects
6. To conduct activities necessary to successfully complete and close the Software projects
7. To develop the skills for tracking and controlling software deliverables
8. To create project plans that address real-world management challenges

## Syllabus:

### Unit I: Introduction

Project, Management, Software Project Management activities, Challenges in software projects, Stakeholders, Objectives & goals

Project Planning: Step-wise planning, Project Scope, Project Products & deliverables, Project activities, Effort estimation, Infrastructure

### Unit II: Project Approach

Lifecycle models, Choosing Technology, Prototyping

Iterative & incremental Process Framework: Lifecycle phases, Process Artifacts, Process workflows (Book 2)

### Unit III: Effort estimation & activity Planning

Estimation techniques, Function Point analysis, SLOC, COCOMO, Usecase-based estimation , Activity Identification Approaches, Network planning models, Critical path analysis

### Unit IV: Risk Management

Risk categories, Identification, Assessment, Planning and management, PERT technique, Monte Carlo approach

### Unit V: Project Monitoring & Control , Resource Allocation

Creating a framework for monitoring & control, Progress monitoring, Cost monitoring, Earned value Analysis, Defects Tracking, Issues Tracking, Status reports, Types of Resources, Identifying resource requirements, Resource scheduling

### Unit VI: Software Quality

Planning Quality, Defining Quality - ISO 9016, Quality Measures, Quantitative Quality Management Planning, Product Quality & Process Quality

Metrics, Statistical Process Control Capability Maturity Model, Enhancing software Quality ( Book3)

**Text Books:**

1. Software Project Management, Bob Hughes & Mike Cotterell, TATA Mcgraw-Hill
2. Software Project Management, Walker Royce: Pearson Education, 2005.
3. Software Project Management in practice, Pankaj Jalote, Pearson.

**Reference Book:**

1. Software Project Management, Joel Henry, Pearson Education.

# Machine Learning

## Course objectives:

The main objective of this course is for the students to achieve basic knowledge of artificial intelligence, a deepened technical understanding of machine learning research and theories, as well as practical experience of the use and design of machine learning and data mining algorithms for applications and experiments. The course has a strong focus towards applied IT. The student not only learns how to critically review and compare different algorithms and methods, but how to plan, design, and implement learning components and applications and how to conduct machine learning experiments.

## Course outcomes:

- The student will be able evaluate and compare the performance or, other qualities, of algorithms for typical learning problems.
- The student will be able to design a supervised or unsupervised learning system.

## Syllabus:

### UNIT I: Introduction :

Well-posed learning problems, Designing a learning system, Perspectives and issues in machine learning. Concept learning and the general to specific ordering – Introduction, A concept learning task, Concept learning as search, Find-S: finding a maximally specific hypothesis, Version spaces and the candidate elimination algorithm, Remarks on version spaces and candidate elimination, Inductive bias.

### UNIT II: Linear Regression & Logistic Regression:

**Predicting numeric values: regression** - Finding the best fit lines with linear regression, Locally weighted linear regression, Shrinking Coefficients, The bias / Variance tradeoff.

**Logistic Regression:** Classification with logistic regression and the sigmoid function, Using optimization to find the best regression coefficients.

### UNIT III: Artificial Neural Networks:

Introduction, Neural network representation, Appropriate problems for neural network learning, Perceptions, Multilayer networks and the back propagation algorithm, Remarks on the back propagation algorithm, An illustrative example face recognition, Advanced topics in artificial neural networks

**UNIT IV: Evaluation Hypotheses:** Motivation, Estimation hypothesis accuracy, Basics of sampling theory, A general approach for deriving confidence intervals, Difference in error of two hypotheses, Comparing learning algorithms.

### UNIT V: Support vector machines & Dimensionality Reduction techniques:

Separating data with the maximum margin, finding the maximum margin, efficient optimization with SMO algorithm, speeding up optimization with full platt SMO, Using Kernels for more Complex data.

**Dimensionality Reduction techniques:** Principal Component analysis, Example.

### UNIT VI:

**Instance-Based Learning-** Introduction, k -Nearest Neighbor Learning, Locally Weighted Regression, Radial Basis Functions, Case-Based Reasoning, Remarks on Lazy and Eager Learning.

**Genetic Algorithms:** Representing Hypotheses, Genetic Operators, Fitness Function and Selection, Illustrative Example.

## TEXT BOOKS:

1. Machine Learning ,Tom M. Mitchell, MGH
2. Machine Learning in Action, Peter Harington, 2012, Cengage.

## REFERENCE BOOKS:

1. Introduction to Machine Learning, Ethem Alpaydin, PHI, 2004

# Advanced Databases

## Course Objectives:

1. Be able to design high-quality relational databases and database applications.
2. Have developed skills in advanced visual & conceptual modeling and database design.
3. Be able to translate complex conceptual data models into logical and physical data
4. Base designs.
5. Have developed an appreciation of emerging database trends as they apply to semi-structured data, the internet, and object-oriented databases

## Course Outcomes:

1. Identify, describe, and categorize database objects
2. Design and implement advanced queries using Structured Query Language
3. Design, construct and maintain a database and various database objects using procedural language constructs, forms and reports to solve problems
4. Administer a database by recommending and implementing procedures including database tuning, backup and recovery
5. Propose, implement and maintain database security mechanisms
6. Explore non-relational database systems and structures

## Syllabus:

### UNIT – I :

**Algorithms for Query Processing and Optimization:** Translating SQL queries into relational algebra- algorithms for external sorting- algorithms for select and join operations- algorithms for project and set operations- implementing aggregate operations and outer joins- combining operations using pipelining- using heuristics in query optimization.

### UNIT –II:

**Data base systems architecture and the system Catalog:** System architectures for DBMSs, Catalogs for Relational DBMSs, System catalog information in oracle.

**Practical database design and tuning:** Physical Database Design in Relational Databases- an overview of Database Tuning in Relational systems.

### UNIT – III:

**Distributed DBMS Concepts and Design:** Introduction- function and architecture of a Distributed DBMS- Distributed Relational Database Design- transparencies in a Distributed DBMS- Date's Twelve Rules for Distributed DBMS.

**Distributed DBMS-Advanced Concepts:** Distributed Transaction Management- Distributed Concurrency Control- Distributed Deadlock Management- Distributed Database Recovery- The X/Open Distributed Transaction processing model- Replication Servers.

### UNIT – IV:

**Introduction to Object DBMSs:** Advanced Database Applications- Weaknesses of RDBMSs- Object oriented Concepts- Storing objects in a Relational Database- Next generation Database systems.

**Object-Oriented DBMSs- Concepts and Design :** Introduction to Object-Oriented Data Models and DBMSs- OODBMS perspectives- Persistence- Issues in OODBMSs- The object Oriented Database System Manifesto- Advantages and Disadvantages of OODBMSs- Object oriented Database Design.

### UNIT V:

**Object-Oriented DBMSs- Standards and Systems:** Object management group- Object Database Standard ODMG3.0, 1999- Object store.

**Object relational DBMSs:**Introduction to Object-relational Database systems- third generation Database manifesto-Postgres-an early ORDBMS-SQL3.

**UNIT – VI :**

**Emerging database technologies and applications:** Hadoop, Big Data characteristics, NO SQL databases, BASE, Brewer's theorem, Relationship between CAP, ACID and No SQL databases, comparison with Relational databases, No SQL databases types, Comparative study of NoSQL products, Case studies using MangoDB and Cassandra

**TEXT BOOK:**

1. “Fundamentals of Database Systems”, ElmasriNavate, 5/e, Pearson Education.
2. Principles of distributed databases S Ceri and Palgettgi TMH
3. Getting started with No SQL Databases , Gaurav Vaish

**REFERENCES BOOKS:**

1. “Principles of Distributed Database Systems”, Ozsu, 2/e, PHI.

## UML & Design Patterns Lab

**(Textbook no.2 i.e.** Object-Oriented Analysis & Design with the Unified Process by Satzinger, Jackson & Burd Cengage Learning will be the primary source for finding templates for developing different artifacts / diagrams)

### **Take three case studies:**

- **Customer Support System (in the Object-Oriented Analysis & Design with the Unified Process by Satzinger, Jackson & Burd Cengage Learning )**
- **Point-Of-Sale Terminal (in Larman textbook)**
- **Library Management System (in the reference book no. 2 i.e. UML toolkit)**

### **Week 1:**

**Familiarization with Rational Rose or Umbrello**

**For each case study:**

### **Week 2, 3 & 4:**

**For each case study:**

- a) Identify and analyze events
- b) Identify Use cases
- c) Develop event table
- d) Identify & analyze domain classes
- e) Represent use cases and a domain class diagram using Rational Rose
- f) Develop CRUD matrix to represent relationships between use cases and problem domain classes

### **Week 5 & 6:**

**For each case study:**

- a) Develop Use case diagrams
- b) Develop elaborate Use case descriptions & scenarios
- c) Develop prototypes (without functionality)
- d) Develop system sequence diagrams

### **Week 7, 8, 9 & 10:**

**For each case study:**

- a) Develop high-level sequence diagrams for each use case
- b) Identify MVC classes / objects for each use case
- c) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the three-layer objects
- d) Develop detailed design class model (use GRASP patterns for responsibility assignment)
- e) Develop three-layer package diagrams for each case study

### **Week 11 & 12:**

**For each case study:**

- a) Develop Use case Packages
- b) Develop component diagrams
- c) Identify relationships between use cases and represent them
- d) Refine domain class model by showing all the associations among classes

### **Week 13 onwards:**

**For each case study:**

- a) Develop sample diagrams for other UML diagrams - state chart diagrams, activity diagrams and deployment diagrams



**Mobile Application Development Lab**

1. Write a J2ME program to show how to change the font size and colour.
2. Write a J2ME program which creates the following kind of menu.
  - \* cut
  - \* copy
  - \* past
  - \* delete
  - \* select all
  - \* unselect all
3. Create a J2ME menu which has the following options (Event Handling):
  - cut - can be on/off
  - copy - can be on/off
  - paste - can be on/off
  - delete - can be on/off
  - select all - put all 4 options on
  - unselect all - put all
4. Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array. You can enter four data (integer) values to the input text field.
5. Create an MIDP application which examine, that a phone number, which a user has entered is in the given format (Input checking):
  - \* Area code should be one of the following: 040, 041, 050, 0400, 044
  - \* There should 6-8 numbers in telephone number (+ area code)
6. Write a sample program to show how to make a SOCKET Connection from J2ME phone. This J2ME sample program shows how to how to make a SOCKET Connection from a J2ME Phone. Many a times there is a need to connect backend HTTP server from the J2ME application. Show how to make a SOCKET connection from the phone to port 80.
7. Login to HTTP Server from a J2ME Program. This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server. Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server. Note: Use Apache Tomcat Server as Web Server and MySQL as Database Server.
8. The following should be carried out with respect to the given set of application domains: (Assume that the Server is connected to the well-maintained database of the given domain. Mobile Client is to be connected to the Server and fetch the required data value/information)
  - Students Marks Enquiry
  - Town/City Movie Enquiry
  - Railway/Road/Air (For example PNR) Enquiry/Status
  - Sports (say, Cricket) Update
  - Town/City Weather Update
  - Public Exams (say Intermediate or SSC)/ Entrance (Say EAMCET) Results EnquiryDivide Student into Batches and suggest them to design database according to their domains and render information according the requests.
9. Write an Android application program that displays Hello World using Terminal.
10. Write an Android application program that displays Hello World using Eclipse.

11. Write an Android application program that accepts a name from the user and displays the hello name to the user in response as output using Eclipse.

12. Write an Android application program that demonstrates the following:

- (i) LinearLayout
- (ii) RelativeLayout
- (iii) TableLayout
- (iv) GridView layout

13. Write an Android application program that converts the temperature in Celsius to Fahrenheit.

14. Write an Android application program that demonstrates intent in mobile application development.

**Software Testing Lab****Lab Assignments****Problem Statement 01**

Consider an automated banking application. The user can dial the bank from a personal computer, provide a six-digit password, and follow with a series of keyword commands that activate the banking function. The software for the application accepts data in the following form:

Area Code	Blank or three-digit number
Prefix	Three-digit number, not beginning with 0 or 1
Suffix	Four-digit number
Password	Six-character alphanumeric
Commands	"Check status", "Deposit", "Withdrawal"

Design adhoc test cases to test the system

**Problem Statement 02**

Consider an automated banking application. The user can dial the bank from a personal computer, provide a six-digit password, and follow with a series of keyword commands that activate the banking function. The software for the application accepts data in the following form:

Area Code	Blank or three-digit number
Prefix	Three-digit number, not beginning with 0 or 1
Suffix	Four-digit number
Password	Six-character alphanumeric
Commands	"Check status", "Deposit", "Withdrawal"

Design the test cases to test the system using following Black Box testing technique:

- BVA, Worst BVA, Robust BVA, Robust Worst BVA
- Equivalence class testing (Input/Output domain)

**Problem Statement 03**

Consider an application that is required to validate a number according to the following simple rules:

1. A number can start with an optional sign.
2. The optional sign can be followed by any number of digits.
3. The digits can be optionally followed by a decimal point, represented by a period.
4. If there is a decimal point, then there should be two digits after the decimal.
5. Any number-whether or not it has a decimal point, should be terminated a blank.
6. A number can start with an optional sign.
7. The optional sign can be followed by any number of digits.
8. The digits can be optionally followed by a decimal point, represented by a period.
9. If there is a decimal point, then there should be two digits after the decimal.
10. Any number-whether or not it has a decimal point, should be terminated a blank. Generate test cases to test valid and invalid numbers.

**(HINT)** Use Decision table and cause-effect graph to generate test cases.

#### Problem Statement 04

Generate test cases using Black box testing technique to Calculate Standard Deduction on Taxable Income. The standard deduction is higher for tax payers who are 65 or older or blind. Use the method given below to calculate tax.

1. The first factor that determines the standard deduction is the filing status. The basic standard deduction for the various filing status are:

Single	\$4,750
Married, filing a joint return	\$9,500
Married, filing a separate return	\$7,000

2. If a married couple is filing separate returns and one spouse is not taking standard Deduction, the other spouse also is not eligible for standard deduction.

3. An additional \$1,000 is allowed as standard deduction, if either the filer is 65 yrs or the spouse is 65 yrs or older (the latter case applicable when the filing status is "Married" and filing "joint").

4. An additional \$1,000 is allowed as standard deduction, if either the filer is blind or the spouse is blind (the latter case applicable when the filing status is "married" and filing "joint").

#### (HINT):

From the above description, it is clear that the calculation of standard deduction depends on the following 3 factors:

1. Status of filing of the filer
2. Age of the filer
3. Whether the filer is blind or not

In addition, in certain cases, the following additional factors also come into play in calculating the standard deduction.

1. Whether spouse has claimed standard deduction
2. Whether spouse is blind
3. Whether the spouse is more than 65 years old

#### Problem Statement 05

Consider the following program segment:

1. int max (int i, int j, int k)
2. {
3. int max;
4. if (i>j) then
5. if (i>k) then max=i;
6. else max=k;
7. else if (j > k) max=j
8. else max=k
9. return (max);
10. }

- a) Draw the control flow graph for this program segment
- b) Determine the cyclomatic complexity for this program
- c) Determine the independent paths

#### Problem Statement 06

Source code of simple insertion sort implementation using array in ascending order in c programming language

```
#include<stdio.h>
int main(){
int i,j,s,temp,a[20];
```

```

Printf ("Enter total elements: "); Scanf ("%d",&s);
printf("Enter %d elements: ",s); for(i=0;i<s;i++) scanf("%d",&a[i]); for(i=1;i<s;i++){
temp=a[i]; j=i-1; while((temp<a[j])&&(j>=0)){ a[j+1]=a[j];
j=j-1;
}
a[j+1]=temp;
}
printf("After sorting: ");
for(i=0;i<s;i++)
printf(" %d",a[i]);
return 0;
}

```

**HINT:** for loop is represented as while loop

- Draw the program graph for given program segment
- Determine the DD path graph
- Determine the independent paths
- Generate the test cases for each independent path

### Problem Statement 07

Consider a system having an FSM for a stack having the following states and transitions:

#### States

Initial: Before creation

Empty: Number of elements = 0

Holding: Number of elements > 0, but less than the maximum capacity

Full: Number elements = maximum

Final: After destruction

Initial to Empty: Create

Empty to Holding, Empty to Full, Holding to Holding, Holding to Full: Add

Empty to Final, Full to Final, Holding to Final: Destroy

Holding to Empty, Full to Holding, Full to Empty: Delete

Design test cases for this FSM using state table-based testing.

### Problem Statement 08

Given the following fragment of code, how many tests are required for 100% decision coverage? Give the test cases.

```

if width > length
then biggest_dimension = width if height > width
then biggest_dimension = height end_if
else if biggest_dimension = length then if height > length
then biggest_dimension = height end_if
end_if end_if

```

**Hint** 04 test cases

### Problem Statement 09

Given the following code, how much minimum number of test cases is required for full statement and branch coverage?

```

read p read q
if p+q > 100
then print "Large" endif
if p > 50
then print "p Large" endif

```

**Hint** 1 test for statement coverage, 2 for branch coverage

### Problem Statement 10

Consider a program to input two numbers and print them in ascending order given below. Find all du paths and identify those du-paths that are not feasible. Also find all dc paths and generate the test cases for all paths (dc paths and non dc paths).

```
#include<stdio.h>
#include<conio.h>
1. void main ()
2. {
3 int a, b, t;
4. Clrscr ();
5. Printf (“Enter first number”);
6. scanf (“%d”,&a);
7. printf(“Enter second number”);
8. scanf(“%d”,&b);
9. if (a<b){
10. t=a;
11a=b;
12 b=t;
13}
14. printf (“%d %d”, a, b);
15 getch ();
}
```

### Problem Statement 11

Consider the above program and generate possible program slices for all variables. Design at least one test case from every slice.

### Problem Statement 12

Consider the code to arrange the nos. in ascending order. Generate the test cases for relational coverage, loop coverage and path testing. Check the adequacy of the test cases through mutation testing and also compute the mutation score for each.

```
i = 0;
n=4; //N-Number of nodes present in the graph
While (i<n-1) do j = i + 1;
While (j<n) do
if A[i]<A[j] then swap (A[i], A[j]); end do;
i=i+1;
end do
```

**Hadoop & BigData Lab****Week 1,2:****1. Implement the following Data structures in Java**

- a) Linked Lists b) Stacks c) Queues d) Set e) Map**

**Week 3, 4:**

- 2. (i) Perform setting up and Installing Hadoop in its three operating modes:  
Standalone,  
Pseudo distributed,  
Fully distributed**

**(ii) Use web based tools to monitor your Hadoop setup.****Week 5:****3. Implement the following file management tasks in Hadoop:**

- Adding files and directories
- Retrieving files
- Deleting files

**Hint:** A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.

**Week 6:****4. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.****Week 7:****5. Write a Map Reduce program that mines weather data.**

Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented.

**Week 8:****6. Implement Matrix Multiplication with Hadoop Map Reduce****Week 9,10:****7. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.****Week 11,12:****8. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes**

**Elective - III****Human Computer Interaction****Course Objectives:**

The main objective is to get student to think constructively and analytically about how to design and evaluate interactive technologies.

**Course Outcomes:**

1. Explain the capabilities of both humans and computers from the viewpoint of human information processing.
2. Describe typical human–computer interaction (HCI) models, styles, and various historic HCI paradigms.
3. Apply an interactive design process and universal design principles to designing HCI systems.
4. Describe and use HCI design principles, standards and guidelines.
5. Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.
6. Discuss tasks and dialogs of relevant HCI systems based on task analysis and dialog design.

**Syllabus:****UNIT I:**

**Introduction:** Usability of Interactive Systems- introduction, usability goals and measures, usability motivations, universal usability, goals for our profession

**Managing Design Processes:** Introduction, Organizational design to support usability, Four pillars of design, development methodologies, Ethnographic observation, Participatory design, Scenario Development, Social impact statement for early design review, legal issues, Usability Testing and Laboratories

**UNIT II:**

**Menu Selection, Form Fill-In and Dialog Boxes:** Introduction, Task- Related Menu Organization, Single menus, Combinations of Multiple Menus, Content Organization, Fast Movement Through Menus, Data entry with Menus: Form Fill-in, dialog Boxes, and alternatives, Audio Menus and menus for Small Displays

**UNIT III:**

**Command and Natural Languages:** Introduction, Command organization Functionality, Strategies and Structure, Naming and Abbreviations, Natural Language in Computing

**Interaction Devices:** Introduction, Keyboards and Keypads, Pointing Devices, Speech and Auditory Interfaces, Displays- Small and large

**UNIT IV:**

**Quality of Service:** Introduction, Models of Response-Time impacts, Expectations and attitudes, User Productivity, Variability in Response Time, Frustrating Experiences

**Balancing Function and Fashion:** Introduction, Error Messages, Nonanthropomorphic Design, Display Design, Web Page Design, Window Design, Color

**UNIT V:**

**User Documentation and Online Help:** Introduction, Online Vs Paper Documentation, Reading from paper Vs from Displays, Shaping the content of the Documentation, Accessing the Documentation, Online tutorials and animated documentation, Online communities for User Assistance, The Development Process

**UNIT VI:**

**Information Search:** Introduction, Searching in Textual Documents and Database Querying, Multimedia Document Searches, Advanced Filtering and Searching Interfaces

**Information Visualization:** Introduction, Data Type by Task Taxonomy, Challenges for Information Visualization



**Text Books:**

1. Designing the User Interface, Strategies for Effective Human Computer Interaction, 5ed, Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven M Jacobs, Pearson
2. The Essential guide to user interface design,2/e, Wilbert O Galitz, Wiley DreamaTech.

**Reference Books:**

1. Human Computer, Interaction Dan R.Olsan, Cengage ,2010.
2. Designing the user interface. 4/e, Ben Shneidermann , PEA.
3. User Interface Design, Soren Lauesen , PEA.
4. Interaction Design PRECE, ROGERS, SHARPS, Wiley.

# Advanced Operating Systems

## Course Objectives:

The aim of this module is to study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems); Hardware and software features that support these systems.

## Course Outcomes:

1. Outline the potential benefits of distributed systems
2. Summarize the major security issues associated with distributed systems along with the
3. range of techniques available for increasing system security

## Syllabus:

### UNIT-I:

**Introduction to Distributed systems:** Goals of distributed system, hardware and software concepts, design issues.

**Communication in Distributed systems:** Layered protocols, ATM networks, the Client - Server model, remote procedure call and group communication.

### UNIT-II:

**Synchronization in Distributed systems:** Clock synchronization, Mutual exclusion, E-tech algorithms, the Bully algorithm, a ring algorithm, atomic transactions,

### UNIT-III:

**Deadlocks:** deadlock in distributed systems, Distributed deadlock prevention, and distributed dead lock detection.

### UNIT-IV:

**Processes:** Processes and Processors in distributed systems: Threads, system models, Processor allocation, Scheduling in distributed system, Fault tolerance and real time distributed systems.

### UNIT-V:

**Distributed file systems:** Distributed file systems design, distributed file system implementation, trends in distributed file systems.

**Distributed shared memory :** What is shared memory, consistency models, page based distributed shared memory, shared variable distributed shared memory, object based DSM.

### UNIT-VI:

**Case study MACH :** Introduction to MACH, process management in MACH, memory management in MACH, communication in MACH, UNIX emulation in MACH. Case study DCE : Introduction to DCE threads, RPC's, Time service, Directory service, security service, Distributed file system.

## TEXT BOOKS:

1. Distributed Operating System - Andrew. S. Tanenbaum, PHI
2. Operating Systems' – Internal and Design Principles Stallings, Fifth Edition–2005, Pearson education/PHI

## REFERENCE BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne  
7<sup>th</sup> Edition, John Wiley.
2. Modern Operating Systems, Andrew S Tanenbaum 2<sup>nd</sup> edition Pearson/PHI

# Mobile Adhoc & Sensor Networks

## Course Objectives:

- 1) To make the student understand the concepts of MOBILE AD HOC NETWORKS (Manets) as well as Wireless Sensor Networks (WSN), their characteristics, novel applications, and technical challenges.
- 2) To understand the issues and solutions of various layers of Manets, namely MAC layer, Network Layer & Transport Layer in Manets and WSN.
- 3) To understand the platforms and protocols used in Manets and WSN.
- 4) To make the student take up further research as part of his higher studies

## Course Outcomes:

- 1) Able to think and develop new applications in Manets and WSN.
- 2) Able to take any new technical issue related to these new thrust areas and come up with a solution(s).
- 3) Able to develop algorithms/protocols for Manets and WSN.

## Syllabus:

### UNIT I :

*Introduction to Ad Hoc Networks:* Characteristics of MANETs, applications of MANETs, and challenges of MANETs.

*Routing in MANETs:* Criteria for classification, Taxonomy of MANET routing algorithms, Topology based routing algorithms, Position based routing algorithms,

### UNIT II:

*Data Transmission:* Broadcast storm problem, Broadcasting, Multicasting and Geocasting

### UNIT III:

*TCP over Ad Hoc:* TCP protocol overview, TCP and MANETs, and Solutions for TCP over Ad hoc

### UNIT IV:

*Basics of Wireless Sensors and Applications:* Applications, Classification of sensor networks, Architecture of sensor networks, Physical layer, MAC layer, Link layer

### UNIT V :

*Data Retrieval in Sensor Networks:* Routing layer, Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs, and Sensor Networks and mobile robots.

### UNIT VI :

*Security:* Security in ad hoc networks, Key management, Secure routing, Cooperation in MANETs, and Intrusion detection systems.

*Sensor Network Platforms and Tools:* Sensor Network Hardware, Berkeley motes, Sensor Network Programming Challenges, Node-Level Software Platforms, TinyOS, NS-2 and TOSSIM.

## Textbook:

1. *Ad hoc and Sensor Networks - Theory and Applications*, by Carlos Cordeiro and Dharma P. Agrawal, World Scientific Publications, March 2006, ISBN 981-256-681-3.
2. *Wireless Sensor Networks: An Information Processing Approach*, Feng Zhao, Leonidas Guibas, Elsevier Science ISBN: 978-1-55860-914-3, (Morgan Kauffman)

# Pattern Recognition

## Course Objectives:

The course is designed to introduce students to theoretical concepts and practical issues associated with pattern recognition

## Course Outcomes:

1. Design systems and algorithms for pattern recognition (signal classification), with focus on sequences of patterns that are analyzed using, e.g., hidden Markov models (HMM),
2. Analyse classification problems probabilistically and estimate classifier performance,
3. Understand and analyse methods for automatic training of classification systems,
4. Apply Maximum-likelihood parameter estimation in relatively complex probabilistic models, such as mixture density models and hidden Markov models,
5. Understand the principles of Bayesian parameter estimation and apply them in relatively simple probabilistic models

## Syllabus:

### UNIT-I:

**Introduction:** Machine perception, pattern recognition example, pattern recognition systems, the Design cycle, learning and adaptation

**Bayesian Decision Theory:** Introduction, continuous features – two categories classifications, minimum error-rate classification-zero-one loss function, classifiers, discriminant functions, and decision surfaces

### UNIT-II:

**Normal density:** Univariate and multivariate density, discriminant functions for the normal Density different cases, Bayes decision theory – discrete features, compound Bayesian decision theory and context

### UNIT-III :

**Maximum likelihood and Bayesian parameter estimation:** Introduction, maximum likelihood Estimation, Bayesian estimation, Bayesian parameter estimation–Gaussian case

### UNIT-IV :

**Un-supervised learning and clustering:** Introduction, mixture densities and identifiability, maximum likelihood estimates, application to normal mixtures, K-means clustering. Data description and clustering – similarity measures, criteria function for clustering

### UNIT-V :

**Pattern recognition using discrete hidden Markov models:** Discrete-time Markov process, Extensions to hidden Markov models, three basic problems of HMMs, types of HMMs

### UNIT-VI :

**Continuous hidden Markov models :**

Continuous observation densities, multiple mixtures per state, speech recognition applications.

## Text Books:

1. Pattern classifications, Richard O. Duda, Peter E. Hart, David G. Stroke. Wiley student edition, Second Edition.
2. Pattern Recognition, An Introduction, V Susheela Devi, M Narsimha Murthy, Universiy Press

## Reference Books:

1. R.C Gonzalez and R.E. Woods, “Digital Image Processing”, Addison Wesley, 1992.
2. Pattern Recognition and Image Analysis – Earl Gose, Richard John baugh, Steve Jost PHI 2004
3. Fundamentals of speech Recognition, Lawerence Rabiner, Biing – Hwang Juang Pearson education.
4. Pattern Recognition, Sergios Theodoridis, Konstantinos Koutroumbas, Academic Press, Elsevier, 4ed,

# Digital Image Processing

## Course Objectives:

To make the students to understand

1. The fundamentals of Computer Graphics and Image Processing
2. The concepts related edge detection, segmentation, morphology and image compression methods.

## Course Outcomes:

1. understanding of digital image processing fundamentals: hardware and software, digitization, enhancement and restoration, encoding, segmentation, feature detection
2. ability to apply image processing techniques in both the spatial and frequency (Fourier) domains
3. Ability To understand (i.e., be able to describe, analyse and reason about) how digital images are represented, manipulated, encoded and processed, with emphasis on algorithm design, implementation and performance evaluation

## SYLLABUS:

### UNIT I:

**Introduction:** Applications of Computer Graphics and Image Processing, Fundamentals on Pixel concepts, effect of Aliasing and Jaggles, Advantages of high resolution systems

**DDA line algorithms:** Bresenham's line and circle derivations and algorithms

### UNIT II:

**2-D Transformations:** Translations, Scaling, rotation, reflection and shear transformations, Homogeneous coordinates, **Composite Transformations-** Reflection about an arbitrary line; Windowing and clipping, viewing transformations, Cohen- Sutherland clipping algorithm

### UNIT III:

**Digital Image Properties:** Metric and topological properties of Digital Images, Histogram, entropy, Visual Perception, Image Quality, Color perceived by humans, Color Spaces, Palette Images, color Constancy

**Color Images:** Pixel brightness transformations, Local Preprocessing, image smoothing, Edge detectors, Robert Operators, Laplace, Prewitt, Sobel, Fri-chen, Canny Edge detection

### UNIT IV:

**Mathematical Morphology:** Basic Mathematical Concepts, Binary dilation and Erosion, Opening and closing, Gray Scale dilation and erosion, Skeleton, Thinning , Thickening Ultimate erosion, Geodesic transformations, Morphology and reconstruction, Morphological Segmentation

### UNIT V:

**SEGMENTATION:** Threshold detection methods, Optimal Thresholding, Edge based Segmentation-Edge image thresholding, Edge relaxation, Border tracing, Hough Transforms, Region based segmentation: Region Mergingm Region Splitting, Splitting and Merging, Watershed Segmentation.

### UNIT VI:

**Image Data Compression:** Image data Properties, Discrete Image Transformations in data compression, Discrete Cosine and Wavelet Transforms, Types of DWT and merits; Predictive Compression methods, Hierarchical and Progressive Compression methods, Comparison of Compression methods, JPEG- MPEG Image Compression methods.

## Text Books:

1. Computer Graphics C Version, Donald Hearn, M Paulli Baker , Pearson ( Unit I and Unit II)
2. Image Processing, Analysis and Machine Vision, Millan Sonka, Vaclov Halvoc, Roger Boyle, Cengage Learning, 3ed, ( Unit III, Unit IV, Unit V and Unit VI)

## References:

1. Computer & Machine Vision, Theory , Algorithms , Practicles, E R Davies, Elsevier, 4ed
2. Digital Image Processing with MATLAB and LABVIEW, Vipul Singh, Elsevier
3. Digital Image Processing, R C Gonzalez &R E woods, Addison Pearson, 3ed.

# Microprocessors and Multi core systems

## Course objectives:

The objective of this course is to provide extensive knowledge of microprocessor based systems. The student will

- learn how the hardware and software components of a microprocessor-based system work together to implement system-level features;
- learn both hardware and software aspects of integrating digital devices (such as memory and I/O interfaces) into microprocessor-based systems;

## Course Outcomes:

1. able to solve basic binary math operations using the microprocessor.
2. able to demonstrate programming proficiency using the various addressing modes and data transfer instructions of the target microprocessor.
3. able to program using the capabilities of the stack, the program counter, and the status register and show how these are used to execute a machine code program.
4. able to apply knowledge of the microprocessor's internal registers and operations by use of a PC based microprocessor simulator.
5. able to write assemble assembly language programs, assemble into machine a cross assembler utility and download and run their program on the training boards.

## Syllabus:

### UNIT-I:

overview of microcomputer structure and operation., execution of a three instruction program, microprocessor evolution and types, the 8086 micro processor family , 8086 internal architecture , introduction to programming the 8086,

**8086 family assembly language programming** :Program development steps , constructing the machine codes for 8086 instructions, writing programs for use with an assembler, assembly language program development tools.

( Text Book 1)

### UNIT-II:

#### Implementing standard program structures in 8086 assembly language

Simple sequence programs, jumps, flags and conditional jumps, if-then, if-then-else and multiple if-then-else programs, while-do programs, repeat-until programs, instruction timing and delay loops.

( Text Book 1)

### UNIT-III:

#### Strings, procedures and macros

The 8086 string instructions, writing and using procedures, writing and using assembler macros.

#### 8086 instruction descriptions and assembler directives

Instruction descriptions, assembler directives , DB, DD, DQ, DT, DW, end-program, endp, ends, equ ,even-align on even memory address, extrn , global, public / extrn, group, include, label, length- not implemented IBM MASM, name – off set, ORG, proc, ptr, segment, short, type

( Text Book 1)

### UNIT-IV:

#### 8086: 8086 interrupts and interrupt applications

8086 interrupts and interrupt responses, hardware interrupt applications, Software Interrupts, priority of interrupts, software interrupt applications, programming.

**8086 assembly language programmes** - Bit & Logic operations, strings, procedures, Macros, Number Format, Conversions, ASCII operations, signed Numbers Arithmetic, Programming using High level language constructs.

( Text Book 1)

### UNIT-V:

CPU: architecture of Intel 80286 CPU, Intel 80386, and 32-bit CPU- 80486-Microprocessor( No instruction set).( Text Book 2)

### UNIT-VI:

**The Pentium Family and Core 2 Microprocessors:**

Introduction to the Pentium Processor, Pentium II Microprocessor, Pentium III, Pentium IV and Core2 Processors.(  
Text Book 2)

**TEXT BOOKS:**

1. Microprocessors and Interfacing, Douglas V Hall, Revised 2<sup>nd</sup> ed, TMH
2. The Intel Microprocessors, Architecture, programming and interfacing, 8ed, Barry Bray, Pearson
3. The X86 Microprocessors, architecture, Programming and Interfacing(8086 to Pentium), Lyla B Das, PEA

**REFERENCE BOOKS:**

1. The 8088 and 8086 Microprocessors, Programming, Interfacing, Hardware and Applications, Walter A Triebel, Avtar Singh, 4 ed, Pearson

## Elective - IV

### Embedded and Real Time Systems

**Course Objectives:**

Develop an understanding of the technologies behind the embedded computing systems

1. technology capabilities and limitations of the hardware, software components
2. methods to evaluate design tradeoffs between different technology choices.
3. design methodologies

**Course Outcomes:**

Understand the basics of an embedded system

1. Program an embedded system
2. Design, implement and test an embedded system.

Identify the unique characteristics of real-time systems

1. Explain the general structure of a real-time system
2. Define the unique design problems and challenges of real-time systems

**Syllabus:****Unit-I:**

**Introduction to Embedded systems:** What is an embedded system Vs. General computing system, history, classification, major application areas, and purpose of embedded systems. Core of embedded system, memory, sensors and actuators, communication interface, embedded firmware, other system components, PCB and passive components.

**UNIT-II:**

**8-bit microcontrollers architecture:** Characteristics, quality attributes application specific, domain specific, embedded systems. Factors to be considered in selecting a controller, 8051 architecture, memory organization, registers, oscillator unit, ports, source current, sinking current, design examples.

**UNIT-III:**

RTOS and Scheduling, Operating basics, types, RTOS, tasks, process and threads, multiprocessing and multitasking, types of multitasking, non preemptive, preemptive scheduling.

**UNIT-IV:**

Task communication of RTOS, Shared memory, pipes, memory mapped objects, message passing, message queue, mailbox, signaling, RPC and sockets, task communication/synchronization issues, racing, deadlock, live lock, the dining philosopher's problem.

**UNIT-V:**

The producer-consumer problem, Reader writers problem, Priority Inversion, Priority ceiling, Task Synchronization techniques, busy waiting, sleep and waken, semaphore, mutex, critical section objects, events, device, device drivers, how to clause an RTOS, Integration and testing of embedded hardware and fire ware.

**UNIT-VI:**

Simulators, emulators, Debuggers, Embedded Product Development life cycle (EDLC), Trends in embedded Industry, Introduction to ARM family of processor.

**TEXT BOOK:**

1. Introduction to embedded systems Shibu. K.V, TMH, 2009.

**REFERENCE BOOKS:**

1. Ayala & Gadre: The 8051 Microcontroller & Embedded Systems using Assembly and C, CENGAGE



2. Embedded Systems, Rajkamal, TMH, 2009.
3. Embedded Software Primer, David Simon, Pearson.
4. The 8051 Microcontroller and Embedded Systems, Mazidi, Mazidi, Pearson,.

# Neural Networks & Soft Computing

## Course Objectives:

1. To have a detailed study of neural networks, Fuzzy Logic and uses of Heuristics based on human experience.
2. To Familiarize with Soft computing concepts.
3. To introduce the concepts of genetic algorithm and its applications to soft computing using some applications

## Course Outcomes:

1. Identify and describe soft computing techniques and their roles in building intelligent machines.
2. Recognize the feasibility of applying a soft computing methodology for a particular problem.
3. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems.

## Syllabus:

### UNIT I:

**INTRODUCTION:** what is a neural network? Human Brain, Models of a Neuron, Neural networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural

### UNIT II:

**LEARNING PROCESS:** Error Correction learning, Memory based learning, Hebbian learning, Competitive, Boltzmann learning, Credit Assignment Problem, Memory, Adaption, Statistical nature of the learning process.

### UNIT III:

**CLASSICAL & FUZZY SETS:** Introduction to classical sets – properties, operations and relations; Fuzzy sets – memberships, uncertainty, operations, properties, fuzzy relations, cardinalities, membership functions.

### UNIT IV:

**FUZZY LOGIC SYSTEM COMPONENTS:** Fuzzification, Membership value assignment, development of rule base and decision making system, Defuzzification to crisp sets, Defuzzification methods

### UNIT V:

**CONCEPT LEARNING:** Introduction, A concept learning task, Concept learning as search, Find-S: finding a maximally specific hypothesis, Version spaces and the candidate elimination algorithm

**DECISION TREE LEARNING:** Introduction, Decision tree representation, Appropriate problems for decision tree learning, The basic decision tree learning algorithm, Hypothesis space search in decision tree learning

### UNIT VI:

**GENETIC ALGORITHMS:** Motivation, Genetic Algorithms, an Illustrative Example, Hypothesis Space Search, Genetic Programming, Models of Evolution and Learning, Parallelizing Genetic Algorithms

## TEXT BOOKS:

1. Neural networks A comprehensive foundations, Simon Hhaykin, Pearson Education 2nd edition 2004
2. Neural Networks, Fuzzy Logic, Genetic Algorithms: Synthesis and Applications by Rajasekharan and Pai, PHI Publications
3. Machine Learning, Tom M. Mitchell, MGH

## References:

# Social Networks and the Semantic Web

## Course Objectives:

This course addresses the issues needed to realize the vision of the *Semantic Web* through the use of *Intelligent Agents*. The objectives are :

- to understand semantic web
- to understand the role of ontology and inference engines in semantic web

## Course Outcomes:

Students will

1. demonstrate knowledge and be able to explain the three different “named” generations of the web.
2. demonstrate the ability to participate materially in projects that develop programs relating to Web applications and the analysis of Web data.
3. be able to understand and analyze key Web applications including search engines and social networking sites.
4. be able to understand and explain the key aspects of Web architecture and why these are important to the continued functioning of the World Wide Web.
5. be able to analyze and explain how technical changes affect the social aspects of Web-based computing.
6. be able to develop “linked data” applications using Semantic Web technologies.

## Syllabus:

### UNIT-I:

**The Semantic web:** Limitations of the current Web, The semantic solution, Development of the Semantic Web, The emergence of the social web.

### UNIT-II:

**Social Network Analysis:** What is network analysis?, Development of Social Network Analysis, Key concepts and measures in network analysis.

**Electronic sources for network analysis:** Electronic discussion networks, Blogs and online communities, Web-based networks.

### UNIT-III:

**Knowledge Representation on the Semantic Web:** Ontologies and their role in the Semantic Web, Ontology languages for the semantic Web.

### UNIT-IV:

**Modeling and Aggregating Social Network Data:** State of the art in network data representation, Ontological representation of Social individuals, Ontological representation of social relationships, Aggregating and reasoning with social network data.

### UNIT-V:

**Developing social semantic applications:** Building Semantic Web applications with social network features, Flink- the social networks of the Semantic Web community, Open academia: distributed, semantic-based publication management.

### UNIT-VI:

**Evaluation of Web-Based Social Network Extraction:** Differences between survey methods and electronic data extraction, context of the empirical study, Data collection, Preparing the data, Optimizing goodness of fit, Comparison across methods and networks, Predicting the goodness of fit, Evaluation through analysis.

## Text Book:

1. Social Networks and the *Semantic Web*, Peter Mika, Springer, 2007.
2. *Semantic Web Technologies*, Trends and Research in *OntologyBased Systems*, J.Davies, Rudi Studer, Paul Warren, John Wiley & Sons.

## Reference Books:

1. Semantic Web and Semantic Web Services –Liyang Lu Chapman and Hall/CRC Publishers,(Taylor & Francis Group)
2. Information Sharing on the semantic Web– Heiner Stuckenschmidt; Frank Van Harmelen, Springer Publications

# Cloud Computing

**Course Objectives:** The student will learn about the cloud environment, building software systems and components that scale to millions of users in modern internet, cloud concepts capabilities across the various cloud service models including IaaS, PaaS, SaaS, and developing cloud based software applications on top of cloud platforms.

## Course Outcomes:

1. Understanding the key dimensions of the challenge of Cloud Computing
2. Assessment of the economics , financial, and technological implications for selecting cloud computing for own organization
3. Assessing the financial, technological, and organizational capacity of employer's for actively initiating and installing cloud-based applications.
4. Assessment of own organizations' needs for capacity building and training in cloud computing-related IT areas

## Syllabus:

### UNIT I: Systems modeling, Clustering and virtualization:

Scalable Computing over the Internet, Technologies for Network based systems, System models for Distributed and Cloud Computing, Software environments for distributed systems and clouds, Performance, Security And Energy Efficiency

### UNIT II: Virtual Machines and Virtualization of Clusters and Data Centers:

Implementation Levels of Virtualization, Virtualization Structures/ Tools and mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data Center Automation.

### UNIT III: Cloud Platform Architecture:

Cloud Computing and service Models, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms, Inter Cloud Resource Management, Cloud Security and Trust Management. Service Oriented Architecture, Message Oriented Middleware.

### UNIT IV: Cloud Programming and Software Environments:

Features of Cloud and Grid Platforms, Parallel & Distributed Programming Paradigms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments.

### UNIT V: Cloud Resource Management and Scheduling:

Policies and Mechanisms for Resource Management Applications of Control Theory to Task Scheduling on a Cloud, Stability of a Two Level Resource Allocation Architecture, Feedback Control Based on Dynamic Thresholds. Coordination of Specialized Autonomic Performance Managers, Resource Bundling, Scheduling Algorithms for Computing Clouds, Fair Queuing, Start Time Fair Queuing, Borrowed Virtual Time, Cloud Scheduling Subject to Deadlines, Scheduling MapReduce Applications Subject to Deadlines.

### UNIT VI:

**Storage Systems:** Evolution of storage technology, storage models, file systems and database, distributed file systems, general parallel file systems. Google file system., Apache Hadoop, BigTable, Megastore, Amazon Simple Storage Service(S3)

**TEXT BOOKS:**

1. Distributed and Cloud Computing, Kai Hwang, Geoffry C. Fox, Jack J. Dongarra MK Elsevier.
2. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
3. Cloud Computing, A Hands on approach, Arshadeep Bahga, Vijay Madiseti, University Press

**REFERNCE BOOK:**

1. Cloud Computing, A Practical Approach, Anthony T Velte, Toby J Velte, Robert Elsenpeter, TMH
2. Mastering Cloud Computing, Foundations and Application Programming, Raj Kumar Buyya, Christen vecctiola, S Tammarai selvi, TMH

## Distributed Systems

### Course Objectives:

1. provides an introduction to the fundamentals of distributed computer systems, assuming the availability of facilities for data transmission, IPC mechanisms in distributed systems, Remote procedure calls.
2. Expose students to current technology used to build architectures to enhance distributed computing infrastructures with various computing principles

### Course Outcomes:

1. Develop a familiarity with distributed file systems.
2. Describe important characteristics of distributed systems and the salient architectural features of such systems.
3. Describe the features and applications of important standard protocols which are used in distributed systems.
4. Gaining practical experience of inter-process communication in a distributed environment

### Syllabus:

#### UNIT-I:

**Characterization of Distributed Systems:** Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges.

**System Models:** Introduction, Architectural Models- Software Layers, System Architecture, Variations, Interface and Objects, Design Requirements for Distributed Architectures, Fundamental Models- Interaction Model, Failure Model, Security Model.

#### UNIT-II:

**Interprocess Communication:** Introduction, The API for the Internet Protocols- The Characteristics of Interprocess communication, Sockets, UDP Datagram Communication, TCP Stream Communication; External Data Representation and Marshalling; Client Server Communication; Group Communication- IP Multicast- an implementation of group communication, Reliability and Ordering of Multicast.

#### UNIT-III:

**Distributed Objects and Remote Invocation:** Introduction, Communication between Distributed Objects- Object Model, Distributed Object Model, Design Issues for RMI, Implementation of RMI, Distributed Garbage Collection; Remote Procedure Call, Events and Notifications, Case Study: JAVA RMI

#### UNIT-IV:

**Operating System Support:** Introduction, The Operating System Layer, Protection, Processes and Threads –Address Space, Creation of a New Process, Threads.

#### UNIT-V:

**Distributed File Systems:** Introduction, File Service Architecture; Peer-to-Peer Systems: Introduction, Napster and its Legacy, Peer-to-Peer Middleware, Routing Overlays.

**Coordination and Agreement:** Introduction, Distributed Mutual Exclusion, Elections, Multicast Communication.

#### UNIT-VI:

**Transactions & Replications:** Introduction, System Model and Group Communication, Concurrency Control in Distributed Transactions, Distributed Dead Locks, Transaction Recovery; Replication-Introduction, Passive (Primary) Replication, Active Replication.

### TEXT BOOKS:

1. Ajay D Kshemkalyani, Mukesh Sigal, “Distributed Computing, Principles, Algorithms and Systems”, Cambridge
2. George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems- Concepts and Design”, Fourth Edition, Pearson Publication

## Management Science

### Unit I

**Introduction to Management:** Concept –nature and importance of Management – Functions of Management – Evaluation of Management thought- Theories of Motivation – Decision making process-Designing organization structure- Principles of organization - Types of organization structure

### Unit II

**Operations Management:** Principles and Types of Management – Work study- Statistical Quality Control- Control charts (P-chart, R-chart, and Cchart) Simple problems- Material Management: Need for Inventory control- EOQ, ABC analysis (simple problems) and Types of ABC analysis (HML, SDE, VED, and FSN analysis)

### Unit III

**Functional Management:** Concept of HRM, HRD and PMIR- Functions of HR Manager- Wage payment plans(Simple Problems) – Job Evaluation and Merit Rating - Marketing Management- Functions of Marketing – Marketing strategies based on product Life Cycle, Channels of distributions.

### Unit IV

**Project Management:** (PERT/CPM): Development of Network – Difference between PERT and CPM Identifying Critical Path- Probability- Project Crashing (Simple Problems)

### Unit V

**Strategic Management:** Vision, Mission, Goals, Strategy – Elements of Corporate Planning Process – Environmental Scanning – SWOT analysis- Steps in Strategy Formulation and Implementation, Generic Strategy alternatives

### Unit VI

**Contemporary Management Practice:** Basic concepts of MIS, MRP, Justin-Time(JIT) system, Total Quality Management(TQM), Six sigma and Capability Maturity Model(CMM) Levies, Supply Chain Management , Enterprise Resource Planning (ERP), Business Process outsourcing (BPO), Business process Re-engineering and Bench Marking, Balanced Score Card.

### Text Books

1. Dr. P. Vijaya Kumar & Dr. N. Appa Rao, '*Management Science*' Cengage, Delhi, 2012.
2. Dr. A. R. Aryasri, '*Management Science*' TMH 2011.

### References

1. Koontz & Weihrich: '*Essentials of management*' TMH 2011
2. Seth & Rastogi: '*Global Management Systems*, Cengage learning , Delhi, 2011
3. Robbins: '*Organizational Behaviour*, Pearson publications, 2011
4. Kanishka Bedi: '*Production & Operations Management*, Oxford



Publications, 2011

5. Philip Kotler & Armstrong: Principles of Marketing, Pearson publications

6. Biswajit Patnaik: Human Resource Management, PHI, 2011

7. Hitt and Vijaya Kumar: Starategic Management, Cengage learning

**Objective:** To familiarize with the process of management and to provide basic insights into select contemporary management practices.

**Codes/ Tables:** Normal Distribution Function Tables need to be permitted into the examination Halls

**IV Year – II SEMESTER**

**Project**

**T P C**  
**0 0 9**