

# **GOVERNMENT ARTS COLLEGE (AUTONOMOUS)**

**KUMBAKONAM 612 002**

Re - accredited With 'A' Grade by NAAC & Affiliated to Bharathidasan University

## **DEPARTMENT OF COMPUTER SCIENCE**

(Effective for those admitted from 2020-2021 onwards)



### **SYLLABI**

**M.Sc., COMPUTER SCIENCE**

# GOVERNMENT ARTS COLLEGE (AUTONOMOUS), KUMBAKONAM

## DEPARTMENT OF COMPUTER SCIENCE

Name of the Programme: **M.SC COMPUTER SCIENCE**

Academic Year of Revision: **2020-2021**

### **Programme Outcome (PO):**

After the completion of this programme, the students will be able to

**PO1:** Apply knowledge of Computer Science in all relevant fields supposed to pursue during employment

**PO2:** Suggest measures to solve challenging problems of the field with augmented ideas and principles

**PO3:** Become Computer scientists, entrepreneurs, Technical administrators, teachers

**PO4:** Get employed in coveted positions of private and public sector establishments

**PO5:** Carry out research in Computer science arena

### **Programme Specific Outcome (PSO):**

After the completion of this programme, the students will be able to

**PSO1:** Face the job market with high skills in the pivoted area

**PSO2:** Gain the ability to understand the abstract concepts and suggest measures to solve problems of real-time nature related to the field

**PSO3:** Acquire deep knowledge in computer science and computational skills so that they can face boldly to qualify in exams like NET/SET/GATE, UPSC and Banking type examinations

**PSO4:** Inculcate Computational Know-how in all aspects of approach and to tune skills thereof

**PSO5:** Pursue research in Computer Science and other applied areas

## Master of Computer Science (M. Sc.,) Course Structure under CBCS

(Applicable to the candidates admitted during the academic year 2020-2021)

**Eligibility: B.Sc. Computer Science** (1<sup>st</sup> Preference), If no incumbents are available in the preferential category, candidates from other streams are considered in preferential order indented here below:

B.C.A., candidates or B.Sc. IT candidates

(or)

B.Sc., Software Development

(or)

B.Sc., Any Science Degree with Mathematics as Allied or Mathematics as one of the subjects in the Degree level

All the above only from a recognized university or an Examination accepted by the syndicate of the affiliating University as equivalent hitherto.

### Syllabus

Sem ester	Code	Course	Course Title	Cre dit	Exa m Hrs	Marks		Tota l
						Int	Ext	
I		CC-I	Mathematical Foundations of Computer Science	4	3	25	75	100
		CC-II	Web Technologies	4	3	25	75	100
		CC-III	Database Administration and Management	4	3	25	75	100
		CC-IV	Advanced Computer architecture	5	3	25	75	100
		CP-I	Web Technologies Lab	4	3	40	60	100
			<b>Total</b>	<b>21</b>	-	-	-	<b>500</b>
II		CC-V	Object Oriented System Development	5	3	25	75	100
		CC-VI	Open Source Technologies (PHP&MySQL)	5	3	25	75	100
		EC-I	(a) Mobile Communication (b) Human computer interaction (c) Digital image processing	5	3	25	75	100
		EC-II	(a)Advanced operating system (b) Distributed operating system (c) Network operating System	5	3	25	75	100
		CP-II	Open Source Technologies Lab	4	3	40	60	100
			<b>Total</b>	<b>24</b>	-	-	-	<b>500</b>

III		CC-VII	Principles of Compiler Design	5	3	25	75	100
		CC-VIII	Data Analytics using Python	5	3	25	75	100
		EC-III	(a) Advanced Software Engineering (b) Software Metrics (c) Software Testing	5	3	25	75	100
		EC-IV	(a) Internet of things (IOT) (b) Cryptography and Network Security (c) Cloud Computing	4	3	25	75	100
		CP-III	Data Analytics using Python Lab	4	3	40	60	100
			<b>Total</b>	<b>23</b>	-	-	-	<b>500</b>
IV		CC-IX	.NET frame work	5	3	25	75	100
		CC-X	Real time Programming and Embedded systems	5	3	25	75	100
		EC-V	(a) Artificial Intelligence & Expert Systems (b) Neural Network & Fuzzy system (c) Parallel Processing Architecture	4	3	25	75	100
		CP-IV	.NET Lab	4	3	40	60	100
		PW	Project Work	4	3	25	75	100
		<b>Total</b>	<b>22</b>	-	-	-	<b>500</b>	
<b>Grand Total</b>			<b>90</b>	-	-	-	<b>2000</b>	

#### PG Course Structure

Course Type	Course	Credits
Core (Theory)	10	47
Core(Practical)	4	16
Core(Project Work)	1	4
Elective	5	23
<b>Total</b>	<b>20</b>	<b>90</b>

**GOVERNMENT ARTS COLLEGE (AUTONOMOUS), KUMBAKONAM.**

Re-accredited with 'A' Grade by NAAC & Affiliated to Bharathidasan University

**M.Sc.(COMPUTER SCIENCE)**

**(Effective for those admitted from 2020-2021 onwards)**

**MATHEMATICAL FOUNDATION FOR COMPUTER SCIENCE**

**OBJECTIVE:** To Obtain Knowledge from the Basic elements and derive the Discrete Digital concepts from the subject

**UNIT I:** Mathematical Logic: Propositions, Connectives, Conditional and Bi conditional Proposition, Equivalence of Proposition, Algebra of Proposition, Truth table construction, Predicate Calculus, Quantifiers. Set Theory: Basic Concept And Notation, Set Operation.

**UNIT II:** Relations: Types of Relations, Some operations on Relations, Properties of Relations, Matrix Representation of Relations. Lattices: Principles of Duality, Properties of Lattices, Lattices as Algebraic Systems, Boolean Algebras, Subalgebra, Boolean Expressions and Boolean functions, Logic Gates Karnaugh Map Method, Functions: Definition, Types of Functions, Composition of Functions, Inverse of a Function, Recursive Functions, Hashing Function.

**UNIT III:** Algebraic Structures: Semi groups and Monoids, Groups, subgroups and Homomorphism. Residue Arithmetic, Fermat's Theorem, Euler's Theorem, Group Codes: Communication model of Error Correction, Parity Checker, Error Recovery in Group Codes.

**UNIT IV:** Graphs Theory: Basic Definition, Degree of a Vertex, Edge, Simple Graph, Matrix Representation of Graphs. Incidence Matrix, Adjacency Matrix, Huffman Adjacency Matrix. Paths, Cycles and connectivity. Graph Traversals: DFS and BFS, and Shortest Path Algorithms, Warshall's Algorithm.

**UNIT V:** Trees : Trees, Spanning Tree, Minimum Spanning Tree, Rooted and Binary Trees, Binary Tree, Tree Traversal, Expression Tree, Kruskal's Algorithm and Prim's Algorithm.

**OUT COMES:** After completing this subject the student will be able to

- Gain Knowledge in mathematical logic and algebraic structure. It plays more role in computer science field in the area of Artificial Intelligence.
- And also Gain knowledge in Graph theory and Trees. It is more useful to develop Graph Algorithm. These Algorithms are used to solve graph theoretical concepts to solve the corresponding computer science application.

**TEXT BOOK:** "Discrete Mathematics with Graph Theory and combinatorics",  
T. Veerarajan, Tata McGraw Hill co.,

**REFERENCE BOOK(s):**

1. "Elements of Discrete Mathematics", C.L. Liu and Mohapatra, TMc.Graw Hill co.,
2. "Discrete Mathematical Structures with Applications to Computer Science", J.P. Tremblay, R.Manohar,Tata McGraw Hill co.,

## **WEB TECHNOLOGIES**

### **Objective:**

This Subject is useful for Making own Web page and how to host own web site on internet. Along with that Students will also learn about the protocols involve in internet technology.

### **UNIT I**

Introduction to Web Technologies: Introduction to Web servers like Apache 1.1, IIS XAMPP(Bundle Server), WAMP(Bundle Server),Handling HTTP Request and Response, installations of above servers, HTML and CSS: HTML 5.0, XHTML, CSS 3.

### **UNIT II**

Java Script: An introduction to JavaScript–JavaScript DOM Model-Date and Objects,- Regular Expressions- Exception Handling-Validation-Built-in objects-Event Handling- DHTML with JavaScript. Servlets: Java Servlet Architecture- Servlet Life Cycle- Form GET and POST actions- Session Handling- Understanding Cookies. Installing and Configuring Apache Tomcat Web Server;- DATABASE CONNECTIVITY: JDBC perspectives, JDBC program example - JSP: Understanding Java Server Pages-JSP Standard Tag Library(JSTL)-Creating HTML forms by embedding JSP code.

### **UNIT III**

Introduction to PHP: The problem with other Technologies (Servelets and JSP), Downloading, installing, configuring PHP, Programming in a Web environment and The anatomy of a PHP Page. Overview of PHP Data types and Concepts: Variables and data types, Operators, Expressions and Statements, Strings, Arrays and Functions. PHP Advanced Concepts: Using Cookies, Using HTTP Headers, Using Sessions, Authenticating users, Using Environment and Configuration variables, Working with Date and Time.

## **UNIT IV**

Creating and Using Forms: Understanding Common Form Issues, GET vs. POST, Validating form input, Working with multiple forms, and Preventing Multiple Submissions of a form. XML: Basic XML- Document Type Definition XML Schema DOM and Presenting XML, XML Parsers and Validation, XSL and XSLT Transformation, News Feed (RSS and ATOM).

## **UNIT V**

AJAX: Ajax Client Server Architecture-XML Http Request Object-Call Back Methods; Web Services: Introduction- Java web services Basics – Creating, Publishing, Testing and Describing a Web services (WSDL)-Consuming a web service, Database Driven web service from an application– SOAP.

### **Outcome:**

Students studying the subject would have the capability to make own web site and host their own web site on internet. Also students would have enough knowledge about what are the technologies used in internet.

### **Text and Reference Books:**

1. Beginning PHP and MySQL, 3rd Edition , Jason Gilmore, Apress Publications (Dream tech.)
2. PHP 5 Recipes A problem Solution Approach Lee Babin, Nathan A Good, Frank M.Kromann and Jon Stephens.
3. Deitel and Deitel and Nieto, —Internet and World Wide Web - How to Program, Prentice Hall, 5 th Edition, 2011.
4. Herbert Schildt, —Java-The Complete Reference, Eighth Edition, Mc Graw Hill Professional, 2011.



## DATABASE ADMINISTRATION AND MANAGEMENT

Course Objectives: The objective of the course is to present an introduction to database management systems with an emphasis on how to organize, maintain and retrieve efficiently and effectively information from a DBMS

### UNIT – I:

**Introduction:** Database-System Applications – Purpose of Database Systems – View of Data – Database Languages – Relational Databases – Database Design – Data Storage and Querying – Data Mining and Analysis – Database Architecture – Database Users and Administrators – History of Database Systems.

### UNIT – II:

**Relational Model:** Structure of Relational Databases – Fundamental Relational Algebra Operations – Additional Relational Algebra Operations – Extended Relational Algebra Operations – Null Values – Modification of the Database. **Storage and File Structure:** File Organization – Organization of Records in Files – Data-Dictionary Storage.

### UNIT – III:

**SQL:** Overview of the SQL Query Language – Data Definition – Basic Structure of SQL Queries – Set Operations – Aggregate Functions – Null Values – Nested Subqueries – Modification of the Database. **Advanced SQL:** SQL Data Types and Schemas – Integrity Constraints – Authorization. **Other Relational Languages:** The Tuple Relational Calculus – The Domain Relational Calculus.

### UNIT – IV:

**Database Design and the E-R Model:** Overview of the Design Process – The Entity Relationship Model – Constraints - Entity-Relationship Diagrams – Weak Entity Sets. **Relational Database Design:** Features of Good Relational Designs – Atomic Domains and First Normal Form – Decomposition Using Functional Dependencies – Functional Dependency Theory – Decomposition Using Functional Dependencies – Decomposition Using Multivalued Dependencies.

### UNIT – V:

**Indexing and Hashing:** Basic Concepts – B+-Tree Index Files – B-Tree Index Files – Static Hashing – Dynamic Hashing. **Transactions:** Transaction Concept – Transaction State – Implementation of Atomicity and Durability. **Concurrency Control:** Lock-Based Protocols – Timestamp-Based Protocols – Validation-Based Protocols. **Distributed Databases:** Homogeneous and Heterogeneous Databases – Distributed Data Storage – Distributed Transactions – Commit Protocols.

Learning Outcomes:

- Describe the fundamental elements of relational database management systems
- Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL

- Design ER-models to represent simple database application scenarios
- Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data
- Improve the database design by normalization
- Familiarize with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing

**Text Book :**

Database System Concepts, Fifth Edition, Abraham Silberschatz, Henry F. Korth and S. Sudarshan, McGraw-Hill – 2006.

**References:**

1. Database Systems: Models, Languages, Design and Application, Ramez Elmasri, Pearson Education 2014.
2. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, McGraw Hill, Third Edition 2004.

## ADVANCED COMPUTER ARCHITECTURE

### Objective:

- To know the number system
- To know about logic Gates and its system
- To know about combinational and sequential circuits functioning
- To know about Register Functioning
- To know about Data transfer and Data Flow machine in computers

### Unit I

**Number Systems:** Decimal, Binary, Octal and Hexadecimal Systems – Conversion from one system to another – Binary Addition, Subtraction, Multiplication and Division. **Basic Logic Gates:** Primary Gates, Universal Logic – Boolean Laws and Theorems – Boolean Expressions

### Unit II

**Combinational Circuits and Sequential Circuits::** Multiplexers – Demultiplexers – Decoders – Encoders – Arithmetic Building Blocks – Half and Full Adders – Half and Full Subtractors – Parallel adder – 2's Complement Adder – Subtractor – BCD Adder. Flip Flops – RS, Clocked RS, D, JK, T and Master-Slave Flip Flops – Shift Register – Counters – Asynchronous, MOD-n and Synchronous Counters – BCD counter – Ring Counter

### Unit III

**Central Processing Unit:** General Register Organization – Stack Organization – Instruction Formats – Addressing Modes – Data Transfer and Manipulation – Program Control – Reduced Instruction Set Computer – CISC characteristics – RISC Characteristics. Data Flow Machines

### Unit IV

**Memory:** Location, capacity, Unit of Transfer, Main memory –RAM, ROM, Cache, Simple Direct Mapping Cache, Associative Mapping, 2 – Way Associative Mapping, Virtual Memory, Physical Memory, Secondary Memory – Hard Disk Functions, Vesa BUS Architecture, Storage Area

### Unit V

**Processor Architecture and peripherals:** Processor Organization, Address Bus, Data Bus, Control Bus, Data Flow, Loop Buffer, Multiple Streams. ASCII Functioning, Keyboard Architecture, Frame Buffer, Screen and Printer Architecture.

### COURSE OUTCOME:

- To understand the basic facts of computing and it's Architecture
- Derivate the conversion of Numbers from Binary to Base Thoughts
- To understand the Data Transfer of Data Flow Machine in Computers.

### Text Books

1. Donald P. Leach, Albert Paul Malvino and Goutam Saha, Digital Principles and Applications, Tata McGraw Hill, Sixth Edition, Third Reprint, 2007, Unit: I, II
2. William Stallings, Computer Organization and Architecture, 2016 Pearson Education

3. Morris Mano M, Computer System Architecture, Prentice Hall of India, Third Edition, 2008

### **Books for Reference**

1. Morris Mano. M, Digital Logic and Computer Design, Prentice Hall of India, 2008.
2. Linda Null, Julia Lobur, The Essentials of Computer Organization and Architecture, fourth Edition 2014.

# WEB TECHNOLOGIES LAB

## Objective:

- To get hands on experience in developing web based applications
  - To design interactive web pages using Scripting languages
  - To learn server side programming
- 
1. Create a simple Web Service that converts the temperature from fahrenheit to celsius and vice versa.
  2. Use the above Web Service on a web page and execute to fetch the results
  3. Create a Web Services provider and make it available on the Internet or intranet.
  4. Create a web based Consumer of an existing web service.
  5. Create a Windows application based consumer of an existing web service.
  6. Write an application that simulates sending a SOAP message as a request and receiving another as a response.
  7. Develop a Web Service that provides images as responses.
  8. Develop a web service that access table contents of a database.
  9. Develop a console based Web Service Client.
  10. Develop a Web intranet/internet based Web Service Client.

## Course Outcome:

- Design simple web pages using markup languages like HTML and XHTML.
- Create dynamic web pages using DHTML and java script that is easy to navigate and use.
- Program server side web pages that have to process request from client side web pages.
- Represent web data using XML and develop web pages.
- Understand various web services and how these web services interact.

## **OBJECT ORIENTED SYSTEM DEVELOPMENT**

### **OBJECTIVE:**

Object oriented systems development is a way to develop software by building self contained modules or objects that can be easily replaced, modified and reused. To understand the Object-based view of systems. To develop robust object-based model -system. To inculcate necessary skills to handle complexity in software design

**UNIT – I:** Introduction: An overview of object oriented systems Development – Object Basics – Object Oriented Systems Development Life Cycle:- Software Development process- Building High Quality Software – Object – Oriented systems development: A use case driven Approach.

**UNIT – II:** Object – oriented methodologies: Rumbaugh methodology – Booch Methodology – Jacobson Methodology – Patterns – Frameworks – The unified Approach – unified modeling language: Introduction to UML Language – UML Diagrams – UML class diagram – use case Diagram – UML Interaction diagram – statechart Diagram – Activity diagram.

**UNIT – III:** Object – oriented Analysis:- use case driven object oriented Analysis : The unified Approach – Use – case model – Object Analysis: classification – Identifying object Relationships, Attributes, and methods – Associations : A – Part – of Relationships – Aggregation – class Responsibility: Identifying Attributes and methods.

**UNIT – IV:** Object oriented Design: Object – oriented Design process and Design Axioms – corollaries – Designing classes: the process – class visibility – Refining attributes – Designing methods and protocols – Packages and managing classes – Access Layer: Database management system – Distributed databases and client – server computing – Multidatabase systems: ODBC – Designing Access Layer classes.

**UNIT – V:** Software Quality Assurance: Introduction – Quality Assurance Tests – testing strategies – Impact of object orientation on Testing –Test cases-Test Plan- System usability and measuring user satisfaction: usability Testing – User Satisfaction Test.

### **Outcomes**

- Ability to analyze and model software specifications.
- Ability to abstract object-based views for generic software systems.
- Ability to deliver robust software components.

### **TEXT BOOK(S):**

1. Ali Bahrami, “Object Oriented Systems Development Using UML” Tata McGraw – Hill. Edition – 2008.
2. Martin Fowler, “UML Distilled”, Second Edition, PHI/Pearson Education.

**REFERENCE BOOK(S):**

1. Stephen R.Shach, "Introduction to Object Oriented Analysis and Design", Tata McGraw-Hill, 2003.
2. James Rumbaugh, Ivar Jacobson, Grady Booch "The Unified Modeling Language Reference Manual" Addison Wesley.
3. Hans-Erik Eriksson, Magnus Penker, Brain Lyons, David Fado, "UML Toolkit" OMG Press Wiley Publishing Inc., 2004.

## **OPEN SOURCE TECHNOLOGIES (PHP & MYSQL)**

**OBJECTIVES:** The objective is to teach the students the principles of open source technology, benefits of open source, and the product that includes permission to use its source code, design documents, or contents. The students will also study and understand the different open source licenses and how to start an open source project.

**UNIT I: OPEN SOURCE :** Introduction : Open Source – Open Source vs. Commercial Software – What is Linux? - Free Software – Where I can use Linux? Linux Kernel – Linux Distributions

**UNIT II: LINUX :** Introduction: Linux Essential Commands - Filesystem Concept - Standard Files - The Linux Security Model - Vi Editor - Partitions creation - Shell Introduction - String Processing - Investigating and Managing Processes - Network Clients - Installing Application

**UNIT III: APACHE :** Introduction - Apache Explained - Starting, Stopping, and Restarting Apache - Modifying the Default Configuration - Securing Apache - Set User and Group - Consider Allowing Access to Local Documentation - Don't Allow public\_html Web sites - Apache control with .htaccess

**UNIT IV: MySQL :** Introduction to MY SQL - The Show Databases and Table - The USE command - Create Database and Tables - Describe Table - Select, Insert, Update, and Delete statement - Some Administrative detail - Table Joins - Loading and Dumping a Database.

**UNIT V: PHP :** PHP Introduction- General Syntactic Characteristics - PHP Scripting - Commenting your code - Primitives, Operations and Expressions - PHP Variables - Operations and Expressions Control Statement - Array - Functions - Basic Form Processing - File and Folder Access - Cookies - Sessions - Database Access with PHP - MySQL - MySQL Functions - Inserting Records - Selecting Records - Deleting Records - Update Records.

**Course Outcome:** On successful completion of this course students should be able

1. To develop android applications.
2. To install and work on Linux.
3. To perform Shell Programming.

### **TEXT BOOK**

1. "Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and PHP", James Lee and Brent Ware, Dorling Kindersley(India) Pvt. Ltd, 2008

### **REFERENCE BOOKS**

1. "Setting Up LAMP: Getting Linux, Apache, MySQL, and PHP and working Together", Eric Rosebrock, Eric Filson, Published by John Wiley and Sons, 2004



## **MOBILE COMMUNICATION**

**OBJECTIVE:** To understand the concepts of Mobile and wireless devices, Mobile IP and WAP.

**UNIT – 1:** Introduction: Applications of mobile Communication- simplified reference model – Multiplexing – Spread spectrum. Medium Access control:- Motivation for specialized MAC – SDMA – FDMA – TDMA – CDMA – Comparison of S/T/F/CDMA Mechanisms – Telecommunications systems: GSM – Mobile services – System Architecture – Protocols – Localization and calling – Handover – Security.

**UNIT – 2:** Satellite systems: Applications– Basics –GEO - LEO-MEO–HEO Routing – Localization – Handover – Broadcast systems: - Cyclical repetition of data Digital audio broadcasting – Digital video broadcasting .

**UNIT – 3:** Wireless LAN: Infra red vs Radio Transmission – Infrastructure and ad-hoc networks – IEEE 802.11 – System Architecture– Protocol Architecture – HiperLAN: - HiperLAN 1 - WATM – BRAN – HiperLAN 2 – Bluetooth Architecture.

**UNIT – 4:** Mobile Network Layer: Mobile IP: Goals, Assumptions and Requirements – Entities and Terminology – IP Packet delivery – Agent discovery – Registration – Tunneling and Encapsulation – Reverse Tunneling – Mobile Ad-hoc Networks: - Routing – Destination sequence distance Vector – Dynamic source Routing – Alternative metrics.

**UNIT – 5:** Mobile Transport Layer:- Traditional TCP: Congestion Control – Slow Start – Fast Retransmit / Fast Recovery – Classical TCP Improvements:- Indirect TCP – Snooping – Transaction – Oriented TCP. WAP: Architecture – Wireless Datagram Protocol – Wireless Transport Layer Security – Wireless Transaction Protocol – Wireless Application Environment – Wireless Markup Language.

**TEXT BOOK(S):**

J.Schiller, Mobile Communication, Addison Wesley, 2000

**REFERENCE BOOK(S)**

1. William C.Y.Lee,- Mobile Communication Design Fundamentals, John Wiley
2. William Stallings-, Wireless Communication and Networks, Person Education
3. WAP – Wireless Application Protocol, Singhal- Pearson Education, 2003.

# Human Computer Interaction

**OBJECTIVES:** Upon successful completion of this course, students should be able to:

- Design, implement and evaluate effective and usable graphical computer interfaces.
- Describe and apply core theories, models and methodologies from the field of HCI.
- Describe and discuss current research in the field of HCI.
- Implement simple graphical user interfaces using the Java Swing toolkit.
- Describe special considerations in designing user interfaces for older adults.

## **Unit I**

**The interaction:** Introduction – Models of interaction – Frameworks and HCI – Ergonomics – Interaction Styles – Elements of WIMP interface – Interactivity – The Context of the interaction  
**- Paradigm:** Introduction – Paradigms for interaction.

## **Unit II**

**Interaction Design basics:** Introduction – what is design? – User focus – Scenarios – Navigation design – Screen design and layout – Interaction and prototyping - **HCI in the software process:** Introduction – The software lifecycle – Usability engineering – Interactive design and prototyping – Design rationale.

## **Unit III**

**Design rules:** Introduction – Principles to support usability – Standards – Guidelines – Golden rules and heuristics – HCI patterns - **ImplementationSupport:** Introduction – Elements of windowing systems – Programming the application – Using toolkits – User interface management systems.

## **Unit IV**

**Evaluation techniques:** What is evaluation – Goals of evaluation – Evaluation through expert analysis – Evaluation through user participation – Choosing an evaluation method - **Universal Design:** Introduction – Universal design principles – Multi-modal interaction – Designing for diversity.

## **Unit V**

**User Support:** Instruction – Requirements of user support – Approaches to user support – Adaptive help system – Designing user support systems.

## **Outcomes**

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

- Explain the capabilities of both humans and computers from the viewpoint of human information processing.
- Describe typical human–computer interaction (HCI) models and styles, as well as various historic HCI paradigms.
- Apply an interactive design process and universal design principles to designing HCI systems.
- Describe and use HCI design principles, standards and guidelines.
- Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.
- Discuss tasks and dialogs of relevant HCI systems based on task analysis and dialog design.
- Analyze and discuss HCI issues in groupware, ubiquitous computing, virtual reality, multimedia, and Word Wide Web-related environments.

**Text Book:** 1. "Human-computer Interaction" - Alan Dix - Pearson Education - 2004.

## **DIGITAL IMAGE PROCESSING**

**OBJECTIVES:** To introduce the student to various Image processing techniques, Image fundamentals and mathematical transforms necessary for Image processing, Image enhancement techniques, Image restoration and Image Compression procedures. And beyond these, the student would study the image segmentation and representation techniques.

### **UNIT I: DIGITAL IMAGE FUNDAMENTALS**

Elements of digital image processing systems – Vidicon and Digital Camera working principles –Elements of visual perception - brightness, contrast, hue, saturation, mach band effect, Color image fundamentals - RGB, HIS models, Image sampling, Quantization, dither, Two-dimensional mathematical preliminaries, 2D transforms - DFT, DCT, KLT, SVD.

### **UNIT II : IMAGE ENHANCEMENT**

Histogram equalization and specification techniques, Noise distributions, Spatial averaging, Directional Smoothing, Median, Geometric mean, Harmonic mean, Contra-harmonic mean filters, Homo-morphed filtering, Color image enhancement

### **UNIT III : IMAGE RESTORATION**

Image Restoration – degradation model, Unconstrained restoration – Lagrange multiplier and Constrained restoration, Inverse filtering- removal of blur caused by uniform linear motion, Wiener filtering, Geometric transformations- spatial transformations.

### **UNIT IV : IMAGE SEGMENTATION**

Edge detection, Edge linking via Hough transform – Thresholding – Region based segmentation – Region growing –Region splitting and Merging – Segmentation by morphological watersheds – basic concepts – Dam construction – Watershed segmentation algorithm.

### **UNIT V: IMAGE COMPRESSION**

Need for data compression – Huffman - Run Length Encoding - Shift codes - Arithmetic coding - Vector Quantization - Transform coding - JPEG standard - MPEG.

### **TEXTBOOK**

1. RafaelC.Gonzalez, RichardE.Woods- 'DigitalImageProcessing', Pearson, SecondEdition,2004
2. AnilK.Jain,,FundamentalsOfDigitalImageProcessing',Pearson2002

### **REFERENCES**

1. Kenneth R.Castleman, Digital Image Processing, Pearson, 2006.
2. Rafael C.Gonzalez, Richard E.Woods, Steven Eddins,'Digital Image Processing using MATLAB', PearsonEducation, Inc., 2004
3. D,E.Dudgeon and RM.Mersereau,'Multidimensional Digital Signal Processing', Prentice Hall Professional Technical Reference,1990.
4. William K.Pratt,'Digital Image Processing', John Wiley, New York, 2002
5. Milan Sonkaetal,'IMAGE PROCESSING, ANALYSIS AND MACHINEVISION', Brookes/Cole,Vikas Publishing House, 2<sup>nd</sup> edition,1999.

# ADVANCED OPERATING SYSTEMS

## OBJECTIVES:

### The student should be made to:

- Study the basic concepts and functions of operating systems.
- Understand the structure and functions of OS.
- Learn about Processes, Threads and Scheduling algorithms.
- Understand the principles of concurrency and Deadlocks.
- Learn various memory management schemes.
- Study I/O management and File systems.
- Learn the basics of Linux system and perform administrative tasks on Linux Servers.

Unit 1: OVERVIEW: Introduction: What is Operating Systems - Computer System Organization – Computer System Architecture - Operating System Structure - Operating System Operations . Operating System Structures: Operating System Services - User and Operating - System Interface - System Calls - Types of System Calls.

Unit 2: PROCESS MANAGEMENT: Processes: Process Concept - Process Scheduling. CPU Scheduling: Basic Concepts - Scheduling Criteria. Deadlocks: System Model - Deadlock Characterization - Methods for Handling Deadlocks - Deadlock Prevention - Deadlock Avoidance - Deadlock Detection - Recovery from Deadlock

Unit 3: MEMORY MANAGEMENT: Main Memory: Background - Swapping - Contiguous Memory Allocation – Segmentation. Virtual Memory: Background - Demand Paging.

Unit 4: STORAGE MANAGEMENT: Mass-Storage Structure: Overview of Mass-Storage Structure - Disk Structure - Disk Attachment - Disk Scheduling - Disk Management. File-System Interface: File Concept - Access Methods. File-System Implementation: File System Structure - File System Implementation.

Unit 5: PROTECTION AND SECURITY : Protection : Goals of Protection - Principles of Protection - Domain of Protection. Security : The Security Problem - Program Threats - Firewalling to Protect Systems and Networks – Computer Security Classifications. Case studies: The Linux System - Linux History - Design Principles.

## OUTCOMES:

### At the end of the course, the student should be able to:

- Design various Scheduling algorithms.
- Apply the principles of concurrency.
- Design deadlock, prevention and avoidance algorithms.
- Compare and contrast various memory management schemes.

- Design and Implement a prototype file systems.
- Perform administrative tasks on Linux Servers.

**TEXT BOOK:**

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, 9<sup>th</sup> Edition, John Wiley and Sons Inc., 2012.

**REFERENCES:**

1. William Stallings, “Operating Systems – Internals and Design Principles”, 7<sup>th</sup> Edition, Prentice Hall, 2011.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, Addison Wesley, 2001.
3. Charles Crowley, “Operating Systems: A Design-Oriented Approach”, Tata McGraw Hill Education”, 1996.
4. D M Dhamdhare, “Operating Systems: A Concept-Based Approach”, Second Edition, Tata McGraw-Hill Education, 2007.

# **DISTRIBUTED OPERATING SYSTEM**

Course Objectives:

- To provide hardware and software issues in modern distributed systems.
- To get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.
- To analyze the current popular distributed systems such as peer-to-peer (P2P) systems will also be analyzed.

## **UNIT - I**

Evolution – Models – Popularity - Distributed Operating System – Issues – Distributed Computed Environment - Features of a Good Message Passing – Issues- Synchronization – Buffering – Encoding and Decoding of Message Data – Process Addressing – Failure Handling – Group Communication.

## **UNIT - II**

The RPC Model – Transparency – Implementation – Marshaling - Server Management – Parameter Passing Semantics – Call Semantics – Communication protocols – Exception Handling – Optimization

## **UNIT - III**

Clock Synchronization – Event Ordering – Mutual Exclusion – Deadlock – Election Algorithms - Threads.

## **UNIT – IV**

Distributed Shared Memory : Introduction, General Architecture of DSM System, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Heterogeneous DSM.

## **UNIT - V**

Distributed File Systems: Introduction, Desirable Features of a Good Distributed File System, File Models, File Accessing Models, File-Sharing Semantics, File-Caching Schemes, File Replication, Fault Tolerance.

Course Outcomes

CO1: To provide hardware and software issues in modern distributed systems.

CO2: To get knowledge in distributed architecture, naming, synchronization, consistency and replication, fault tolerance, security, and distributed file systems.

CO3: To analyze the current popular distributed systems such as peer-to-peer (P2P) systems will also be analyzed.

CO4: To know about Shared Memory Techniques.

CO5: Have sufficient knowledge about file access.

CO6: Have knowledge of Synchronization and Deadlock.

## **TEXT BOOKS**

1. Pradeep K. Sinha, "Distributed Operating System Concepts and Design ", PHI, New Delhi, 2007.

## **REFERENCES**

1. Andrew S Tanaenbaum, "Modern Operating System", PHI, New Delhi, 2001
2. D.M. Dhamdhare , 2002, Operating System, Tata McGraw-Hill, New Delhi.
3. A.S. Tanenbaum , Operating Systems: Design and Implementation, Prentice-Hall of India, New Delhi.
4. Nutt, 2005, Operating Systems, 3 rd Edition, Pearson Education, Delhi



# Network Operating Systems

- To understand the fundamentals of Mobile communication systems.
- To understand the different multiplexing scheme.
- To comprehend and apply network layer security protocols, Transport layer security protocols, Web security protocols.
- To understand the wireless network security threats.
- To understand the significance of different layers in mobile system.

## UNIT I - INTRODUCTION

Introduction to wireless, mobile and cellular mobile systems- cellular mobile telephone systems, analog and digital cellular systems- frequency reuse, co-channel interference.

## UNIT II – MAC

Medium access control - MAC, SDMA, FDMA, TDMA, CDMA, Hand offs and dropped calls- initiation of handoff, power difference, mobile assisted cell-site and Intersystem handoff.

## UNIT III – COMMUNICATION SYSTEMS

Mobile Telecommunication standards, GSM, DECT, TETRA, IMT-2000, CTEO, satellite systems – GEO, LEO and MEO, and broadcast systems –Digital audio and video broadcasting, IEEE 802.11, HIPERLAN, Bluetooth, Wireless ATM, WATM services.

## Unit-IV

Intruders, Viruses, Worms, Trojan horses, Distributed Denial-Of-Service (DDoS), Firewalls, IDS, Honey nets, Honey pots.

## Unit-V

Introduction to wireless network security, Risks and Threats of Wireless networks, Wireless LAN Security (WEP, WPA).

**Outcome**

- Understand the concepts of mobile and wireless communications.
- Apply the knowledge gained in exploring, application and protocol development.
- Be able to determine appropriate mechanisms for protecting the network.
- Design a security solution for a given application, system with respect to security of the system

**Text Book**

1. Jochen Sciiiller, "Mobile Communications ", Pearson Education India, 2009.

**Reference Book**

1. Theodore S. Rappaport, "Wireless Communications: Principles and Practice", 2/e, Pearson Education, 2010.
2. William C.Y Lee, "Mobile Cellular Telecommunications ", McGraw Hill International Editions, 1995.

## **OPEN SOURCE TECHNOLOGY LAB**

**Objectives:** To provide fundamental concept of internet, with a view to developing professional software development skills. To expose students to free open source software environment and introduce them to use open source packages.

1. Write a server side php program that display marks, total, grade of student in tabular format by accepting user inputs for name, number and marks from a HTML form.
2. Write a php program that adds products that a selected from a web page to a shopping cart.
3. Write a php program to access the data stored in my sql table.
4. Write a php program interface to create a database and insert a table
  - a) Write a php program using classes to create a table
5. Write a php program to create a directory and read contents from the directory
6. Write a shell program to find the details of an user session
7. Write ashell program to change the extension fo a give file
8. Create a mysql table and execute queries to read, add, remove and modify a record from the table.
9. Create a php program for USER AUTHENTICATION
10. Create a php program for UPLOAD FILE TO SERVER

Course Outcomes: At the end of this course student will:

CO1) Implement various applications using build systems

CO2) Understand the installation of various packages in open source operating systems

CO3) Create simple GUI applications using PHP

CO4) Understand various version control systems.

# **PRINCIPLES OF COMPILER DESIGN**

**Objective:** An introduction to the design and implementation of programming language translators. Theoretical aspects of language design and translation are discussed and practically demonstrated by developing a working compiler.

## **Unit I**

Introduction on the phase of the compiler – Lexical Analysis, Regular Expression, Non deterministic Automata, Deterministic Automata equivalent to NFA's. Minimizing the states of DFA, Implementation of Lexical Analyzer.

## **Unit II**

Syntax Analysis – Context free grammars - Top down Parsing Concepts, Recursive Descent Parsing, Predictive Parsers, Non recursive Predictive Parsing – Bottom Up Parsing, Handle pruning, Shift reduce parsing – Operator Precedence Parsing – Error recovery in Parsing, Parser Generators – YACC. [Omit LR Parsers : Sec 4.7]

## **Unit III**

Intermediate Code Generation: Syntax directed Definitions, Construction of Syntax trees – Top down Translation, Bottom up Evaluation of inherited Attributes, Recursive Evaluators, Assigning Space at Compiler Construction time – Type checking.

## **Unit IV**

Storage Organization : Storage Organization, Storage Allocation Strategies, Parameter Passing, Symbol tables, Dynamic Storage Allocation, Intermediate Languages – Representation of Declarations, Assignment Statement, Boolean Expression, Back patching, Procedure calls.

## **Unit V**

Code Generation and Optimization: Design of the code generators, Runtime storage Management, Basic blocks and flow graphs, Register Allocation and Assignment, DAG representation of Basic blocks, Peephole optimization, Code optimization – The principle sources of optimization, Optimization of basic blocks, Global data flow Analysis, Loop optimizations.

## **OUTCOMES:**

On successful completion of the course students will be able to:

1. Specify and analyse the lexical, syntactic and semantic structures of advanced language features

2. Separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation
3. Write a scanner, parser, and semantic analyser without the aid of automatic generators
4. Turn fully processed source code for a novel language into machine code for a novel computer
5. Describe techniques for intermediate code and machine code optimisation
6. Design the structures and support required for compiling advanced language features.

### **Text Book(s)**

1. Alfred V.Aho, Ravi Sethi, Jeffrey D.Ullman, “Compilers – Principles, Techniques and Tools”, 2007, Pearson Education.
2. Dhamdhere D.M., “Compiler Construction Principles and Practice”, 1981, Macmillan India.

### **References**

1. Reinhard Wilhm, Director Mauser, “Compiler Design”, 1995, Addison Wesley.

## DATA ANALYTICS USING PYTHON

**OBJECTIVE:** To Obtain Knowledge from the Python programming and Data Analytics Using Python.

**UNIT I:** Introduction to Python Programming: Arithmetic operator- values and types. variables, expressions and statements: Assignment Statement - Variable Names – Expressions and Statement –String Operation. Function: Function Calls – Math Function - Adding new Functions - Definitions and Uses - Flow of execution, parameters and arguments -Fruitful functions.

**UNIT II:** Conditionals and Recursion: Floor Division and Modulus - Boolean Expression – Logical operators – Conditional Execution- Alternative Executive- chained conditional - Nested conditionals – Iteration – Strings. Lists, Tuples, Dictionaries: Lists: A List is a Sequence – List are Mutable - Traversing a List - List operations - List slices - List methods - Map, Filter and Reduce – Deleting Elements – List and String – List Arguments – Dictionaries –Tuples – Files – Introduction to packages :Package types – Package Installation.

**UNIT III:** Introduction to Data Analysis: Data Analysis – Knowledge domain of the data Analyst – The Data Analysis Process – Quantitative and Qualitative Data Analysis –Scipy. Numpy Library: Basic Operations – Indexing, Slicing, Iterating – Array Manipulation.

**UNIT IV:** Pandas and Data Visualization: Pandas Data Structure-Series – The Data Frame – The Index Object- Sorting and Ranking – Correlation and Covariance .Pandas Reading and Writing: Csv, Text File, HTML Files, JSON Data-Interacting With database – Matplotlib Architecture – pyplot- Line Chart – Bar Chart –Pie Chart.

**UNIT V:** Machine Learning With SciKit: Supervised and Unsupervised Learning – Training Set and Testing set - Supervised Learning with scikit Learn The Iris Flower Dataset – K-Nearest Neighbors Classifier – Diabetes Dataset – Linear Regression: The Least Square Regression – Support Vector Machine(SVMs).

## **OUT COMES:**

After completing this subject the student will be able to

- Gain programming Knowledge to solve Data science task through this language.
- Gain programming Knowledge to solve Data Analytics task through this Language.
- Gain programming Knowledge to solve Big Data Analytics task through this Language.

## **TEXT BOOK:**

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist``, 2<sup>nd</sup> edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016.
2. "Python Data Analytics", Fabio Nelli, Pub: APress

## M.Sc. - ADVANCED SOFTWARE ENGINEERING

### OBJECTIVES:

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

UNIT 1: **Software Engineering** : Definition – Layers – Management – S/W Engineer – Phases in S/W Engineering – Software Process, Project and Product – S/W Life Cycle Models : Waterfall Model , Prototyping Model, Incremental Model, Spiral Models – S/W Requirements : Types of Requirements – Requirements Engineering Process – Requirements Elicitation – Requirements Validation.

UNIT 2: **Software Design**: Basics of Software Design – Architectural Design – Object-Oriented Design – Software Design Notations – Software Design Reviews – Software Coding: Features of Software Code – Coding Methodology – Programming Practice – Code verification techniques - Coding Tools.

UNIT 3 : **Software Testing**: Software Testing Basics – Software Testing Strategies – Levels of Software Testing: Unit Testing – Integration Testing – System Testing – Acceptance Testing – Testing Techniques : White Box Testing –Black Box Testing – Software Testing Tools- Debugging process – Debugging strategies.

UNIT 4 : **Software Maintenance** : Basics of S/W Maintenance – Types of S/W Maintenance. S/W Planning and Scheduling: Responsibilities of S/W Project Manager – Project Planning – Project Scheduling – Project Staffing – S/W Maintenance models - Techniques for maintenance – S/W Maintenance tools.

UNIT 5 : **Software Cost Estimation** : Basics of Cost Estimation – S/W Cost Estimation Process – Software Estimation Models: COCOMO Model – Software Equation – Expert Judgment. Software Quality : Quality Concepts – S/W Quality Assurance Group – Evaluation of Quality – Advanced Topics: Client/Server S/w Engineering – Web Engineering.

### OUTCOMES:

At the end of this course, the students will be able to:

- Understand the advantages of various Software Development Lifecycle Models
- Gain knowledge on project management approaches as well as cost and schedule estimation strategies
- Perform formal analysis on specifications
- Architect and design using architectural styles and design patterns



- Understand software testing approaches
- Understand the advantages of DevOps practices

**TEXT BOOK:**

1."Software Engineering" Rohit Khurana ,ITL ESL, vikas II Edition,2014.

**REFERENCE BOOKS:**

1."Software Engineering – An Engineering Approach", James F Peters And Witold Pedrycz, John Wiley and Sons, New Delhi,2000.

2."Software Engineering – A Practitioner's Approach", Roger S.Pressman, Mcgraw-Hill InternationalEdition,6<sup>th</sup> Edition,2005.

# SOFTWARE METRICS

## Course Objectives:

- To gain basic knowledge about metrics, measurement theory and related terminologies
- To learn measure the quality level of internal and external attributes of the software product
- To introduce the basics of software reliability and to illustrate how to perform planning, executing and testing for software reliability
- To explore various metrics and models of software reliability
- To compare various models of software reliability based on its application

## UNIT-I : What Is Software Quality:

Quality: Popular Views, Quality Professional Views, Software Quality, Total Quality Management, and Summary. Fundamentals Of Measurement Theory: Definition, Operational Definition, And Measurement, Level Of Measurement, Some Basic Measures, Reliability And Validity, Measurement Errors, Be Careful With Correlation, Criteria For Causality, Summary. Software Quality Metrics Overview: Product Quality Metrics, In Process Quality Metrics, Metrics for Software Maintenance, Examples for Metrics Programs, Collecting software Engineering Data.

## UNIT-II : Applying The Seven Basic Quality Tools In Software Development:

Ishikawa's Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts, Scatter Diagram, Control Chart, Cause, and Effect Diagram. The Rayleigh Model: Reliability Models, the Rayleigh Model Basic Assumptions, Implementation, Reliability and Predictive Validity.

## UNIT-III : Complexity Metrics and Models:

Lines of Code, Halstead's Software Science, Cyclomatic Complexity Syntactic Metrics, An Example of Module Design Metrics in Practice .Metric And Lessons Learned for Object Oriented Projects: Object Oriented Concepts And Constructs, Design And Complexity Metrics, Productivity Metrics, Quality And Quality Management Metrics, Lessons Learned For object oriented Projects.

## UNIT-IV : Availability Metrics:

Definition and Measurement of System Availability, Reliability Availability and Defect Rate, Collecting Customer Outage Data For Quality Improvement, In Process Metrics For Outage And Availability. Conducting Software Project Assessment: Audit Ad Assessment, Software Process Maturity Assessment And Software Project Assessment, Software Process Assessment A Proposed Software Project Assessment Method.

## UNIT-V : Dos And Don'ts Of Software Process Improvement :

Measuring Process Maturity, Measuring Process Capability, Staged Versus Continuous Debating Religion, Measuring Levels Is Not Enough, Establishing The Alignment Principle ,Take Time Getting Faster, Keep it Simple Or Face Decomplexification, Measuring The Value Of Process Improvement ,Measuring Process Compliance , Celebrate The Journey Not Just The Destination. Using Function Point Metrics to Measure Software Process Improvement: Software Process Improvement Sequences, Process Improvement Economies, Measuring Process Improvement at Activity Levels.

**Course Outcomes:**

Upon completion of the course, students shall be able to

- Identify and apply various software metrics, which determines the quality level of software
- Identify and evaluate the quality level of internal and external attributes of the software product
- Compare and Pick out the right reliability model for evaluating the software
- Evaluate the reliability of any given software product
- Design new metrics and reliability models for evaluating the quality level of the software based on the requirement.

**TEXT BOOK:**

1) Norman E-Fentor and Share Lawrence Pflieger.” Software Metrics”. International Thomson Computer Press, 1997.

2) Stephen H Khan: Metrics and Models in Software Quality Engineering, Pearson 2nd edition 2013.

**REFERENCES:**

1) S.A. Kelkar, “Software quality and Testing, PHI Learning, Pvt., Ltd., New Delhi 2012.

2) Watts S Humphrey, “Managing the Software Process”, Pearson Education Inc, 2008.

3) Mary Beth Chrissis, Mike Konrad and Sandy Shrum, “CMMI”, Pearson Education (Singapore) Pvt. Ltd., 2003

# **INTERNET OF THINGS (IOT)**

## **Objective:**

1. Vision and Introduction to IoT.
2. Understand IoT Market perspective.
3. Data and Knowledge Management and use of Devices in IoT Technology.
4. Understand State of the Art – IoT Architecture.

## **Unit I:**

M2M to IoT-The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics.

## **Unit II:**

M2M to IoT – A Market Perspective – Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT - An Architectural Overview – Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

## **Unit III:**

M2M and IoT Technology Fundamentals - Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management.

## **Unit IV:**

IoT Architecture-State of the Art – Introduction, State of the art, Architecture Reference Model-Introduction, Reference Model and architecture, IoT reference Model.

## **Unit V:**

IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints - Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control. Industrial Automation- Service-oriented architecture-based device integration, SOCRADES: realizing the enterprise integrated Web of Things, IMC-AESOP: from the Web of Things to the Cloud of Things, Commercial Building Automation- Introduction, Case study: phase one-commercial building automation today, Case study: phase two-commercial building automation in the future.

## **TEXT BOOK(S):**

1. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatias Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1<sup>st</sup> Edition, Academic Press, 2014.

## **REFERENCE BOOK(S):**

1. Vijay Madiseti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1<sup>st</sup> Edition, VPT, 2014.
2. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1<sup>st</sup> Edition, Apress Publications, 2013.

# CLOUD COMPUTING

## **OBJECTIVES:**

- To understand the concept of cloud computing
- To appreciate the evolution of cloud from existing technologies
- To have knowledge on the various issues in cloud computing
- To be familiar with the lead players in cloud
- To appreciate the emergence of cloud as the next generation computing paradigm.

## **Unit I**

Introduction: History of Centralized and Distributed Computing – Overview of Distributed Computing, Cluster computing, Grid computing. Technologies for Network-based systems- System models for Distributed and cloud computing- Software environments for distributed systems and clouds.

## **Unit II**

Virtualization: Introduction to Cloud Computing- Cloud issues and challenges – Properties – Characteristics – Service models, Deployment models. Cloud resources: Network and API – Virtual and Physical computational resources – Data-storage. Virtualization concepts – Types of Virtualization- Introduction to Various Hypervisors – High Availability (HA)/Disaster Recovery (DR) using Virtualization, Moving VMs.

## **Unit III**

Service Models: Infrastructure as a Service (IaaS) – Resource Virtualization: Server, Storage, Network – Case studies. Platform as a Service (PaaS) – Cloud platform & Management: Computation, Storage – Case studies. Software as a Service (SaaS) – Web services – Web 2.0 – Web OS – Case studies – Anything as a service (XaaS) – Microservices.

## **Unit IV**

Cloud Programming and Software Environments: Cloud Programming and Software Environments – Parallel and Distributed Programming paradigms – Current technologies – Programming support of App Engines – Emerging Cloud software Environment.

## **Unit V**

Cloud Access: authentication, authorization and accounting – Cloud Provenance and meta data – Cloud Reliability and fault-tolerance – Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards.

## **Course Outcomes**

- Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.
- Compare the advantages and disadvantages of various cloud computing platforms.

- c. Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.
- d. Program data intensive parallel applications in the cloud.

## **REFERENCE**

1. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, “Distributed and cloud computing from Parallel Processing to the Internet of Things”, Morgan Kaufmann, Elsevier, 2012.
2. Barrie Sosinsky, “Cloud Computing Bible”, John Wiley & Sons, 2010. 2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance”, O’Reilly, 2009.

## DATA ANALYTICS USING PYTHON LAB

### **Objectives:**

This Lab course will help students to work on python language and gain

1. Gain knowledge to work on Python Programming.
2. Gain Knowledge to solve Data science task through Python Programming.
3. Gain Knowledge to solve Big Data Analytics task through Python Programming.

### **LIST:-**

1. Write a program to perform string operation using python programming.
2. Write a program to perform Function operation using python programming.
3. Write a program to perform List and Tuples using python programming.
4. Write a program to perform Dictionaries using python programming.
5. Write a program using Spicy and Numpy for Data Analytics using python programming.
6. Write a program using Pandas Data Frames for Data Analytics using python programming.
7. Write a program using Pandas Data Frames for Data Analytics using python programming.
8. Write a Program to classify the image using machine Learning by Scikit package using python programming.
9. Write a Program to classify the image using machine Learning(Deep Learning) by Scikit package using python programming.

### **OUT COMES:**

After completed this lab the student will able.

- Develop program or software in the area of Data Science Field .
- Develop program or software in the area of Data Analytics Field .

# **.NET FRAMEWORK**

**OBJECTIVE:** Understand the concept of .NET framework, study the different techniques of security, introduce web services with ASP.NET and explore window based applications.

## **UNIT I: INTRODUCTION TO C#**

Introducing C#, Understanding .NET, overview of C#, Literals, Variables, Data Types, Operators, checked and unchecked operators, Expressions, Branching, Looping, Methods, implicit and explicit casting, Constant, Arrays, Array Class, Array List, String, String Builder, Structure, Enumerations, boxing and unboxing.

## **UNIT II: OBJECT ORIENTED ASPECTS OF C#**

Class, Objects, Constructors and its types, inheritance, properties, indexers, index overloading, polymorphism, sealed class and methods, interface, abstract class, abstract and interface, operator overloading, delegates, events, errors and exception, Threading.

## **UNIT III: APPLICATION DEVELOPMENT ON .NET**

Building windows application, Creating our own window forms with events and controls, menu creation, inheriting window forms, SDI and MDI application, Dialog Box(Modal and Modeless), accessing data with ADO.NET, DataSet, typed dataset, Data Adapter, updating database using stored procedures, SQL Server with ADO.NET, handling exceptions, validating controls, windows application configuration.

## **UNIT IV: WEB BASED APPLICATION DEVELOPMENT ON .NET**

Programming web application with web forms, ASP.NET introduction, working with XML and .NET, Creating Virtual Directory and Web Application, session management techniques, web.config, web services, passing datasets, returning datasets from web services, handling transaction, handling exceptions, returning exceptions from SQL Server.

## **UNIT V: CLR AND .NET FRAMEWORK**

Assemblies, Versioning, Attributes, reflection, viewing meta data, type discovery, reflection on type, marshalling, remoting, security in .NET

### **OUTCOMES:**

After completing this course, the student will be able to

- List the major elements of the .NET frame work
- Explain how C# fits into the .NET platform.
- Analyze the basic structure of a C# application



Debug, compile, and run a simple application.  
Develop programs using C# on .NET  
Design and develop Web based applications on .NET  
Discuss CLR.

**TEXT BOOK:**

1. Herbert Schildt, "The Complete Reference: C# 4.0", Tata McGraw Hill, 2012.
2. Christian Nagel et al. "Professional C# 2012 with .NET 4.5", Wiley India, 2012.

**REFERENCES:**

1. Andrew Troelsen , "Pro C# 2010 and the .NET 4 Platform, Fifth edition, A Press, 2010.
2. Ian Griffiths, Matthew Adams, Jesse Liberty, "Programming C# 4.0", Sixth Edition, O'Reilly, 2010

# **REAL TIME PROGRAMMING AND EMBEDDED SYSTEM**

## **OBJECTIVES:**

- Understand the concepts of embedded system design and analysis
- Learn the architecture and programming of ARM processor
- Be exposed to the basic concepts of embedded programming
- Learn the real time operating systems

## **UNIT I INTRODUCTION TO EMBEDDED SYSTEM DESIGN**

Complex systems and micro processors– Embedded system design process –Design example: Model train controller- Design methodologies- Design flows - Requirement Analysis – Specifications-System analysis and architecture design – Quality Assurance techniques - Designing with computing platforms – consumer electronics architecture – platform-level performance analysis.

## **UNIT II ARM PROCESSOR AND PERIPHERALS**

ARM Architecture Versions – ARM Architecture – Instruction Set – Stacks and Subroutines – Features of the LPC 214X Family – Peripherals – The Timer Unit – Pulse Width Modulation Unit – UART – Block Diagram of ARM9 and ARM Cortex M3 MCU.

## **UNIT III EMBEDDED PROGRAMMING**

Components for embedded programs- Models of programs- Assembly, linking and loading – compilation techniques- Program level performance analysis – Software performance optimization – Program level energy and power analysis and optimization – Analysis and optimization of program size- Program validation and testing.

## **UNIT IV REAL TIME SYSTEMS**

Structure of a Real Time System — Estimating program run times – Task Assignment and Scheduling – Fault Tolerance Techniques – Reliability, Evaluation – Clock Synchronisation.

## **UNIT V PROCESSES AND OPERATING SYSTEMS**

Introduction – Multiple tasks and multiple processes – Multirate systems- Preemptive real- time operating systems- Priority based scheduling- Interprocess communication mechanisms Evaluating operating system performance- power optimization strategies for processes – Example Real time operating systems-POSIX-Windows CE. - Distributed embedded systems MPSoCs and shared memory multiprocessors. – Design Example - Audio player, Engine control unit – Video accelerator.

**OUTCOMES:**

- Describe the architecture and programming of ARM processor
- Outline the concepts of embedded systems
- Explain the basic concepts of real time operating system design
- Model real-time applications using embedded-system concepts

**TEXT BOOKS:**

1. Marilyn Wolf, —Computers as Components - Principles of Embedded Computing System Design, Third Edition —Morgan Kaufmann Publisher (An imprint from Elsevier), 2012. (UNIT I, II, III, V)
2. Jane W.S.Liu, Real Time Systems, Pearson Education, Third Indian Reprint, 2003.(UNIT IV)

**REFERENCES:**

1. Lyla B.Das, —Embedded Systems : An Integrated Approach, Pearson Education, 2013.
2. Jonathan W.Valvano, —Embedded Microcomputer Systems Real Time Interfacing, Third Edition Cengage Learning, 2012.
3. David. E. Simon, —An Embedded Software Primer, 1st Edition, Fifth Impression, Addison-Wesley Professional, 2007.
4. Raymond J.A. Buhr, Donald L.Bailey, —An Introduction to Real-Time Systems- From Design to Networking with C/C++, Prentice Hall, 1999.
5. C.M. Krishna, Kang G. Shin, —Real-Time Systems, International Editions, Mc Graw Hill 1997
6. K.V.K.K.Prasad, —Embedded Real-Time Systems: Concepts, Design & Programming, Dream Tech Press, 2005.
7. Sriram V Iyer, Pankaj Gupta, —Embedded Real Time Systems Programming, Tata Mc Graw Hill, 2004.

## ARTIFICIAL INTELLIGENCE & EXPERT SYSTEMS

### Objective:

- Be able to implement a wide range of AI concepts using Prolog.
- Be able to assess the potential of AI in research and real-world environments.

### UNIT I:

The AI Problems: AI Techniques – Problems, problems space and search – Defining the problem as a state space search – Production systems – Problem Characteristics – Production system characteristics – Heuristic search techniques – Generate and test – Hill climbing – Best first search – Problem reduction – Constraint satisfaction – Means end analysis.

### UNIT II:

Game playing – Mini max search procedure – Adding alpha beta cutoffs– Additional Refinement – Iterative deepening – Using predicate logic – Representing simple facts and Logic – Representing Instance and Isa relationships – Computable functions and predicates – Resolution – Natural deduction.

### UNIT III:

Representing knowledge using rules – Procedural verses declarative knowledge – Logic programming – forward verses backward reasoning – matching – control knowledge – symbolic reasoning under uncertainty – non monotonic reasoning – Logic for non monotonic reasoning – Implementation issues – Augmenting a problem solver – Implementation of depth first search – Implementation of breath first search.

### UNIT IV:

Statistical reasoning – Probability and Bayes's theorem – Certainty factors and Rule based systems – Bayesian Network – Dempster – Shafer theory – Fuzzy logic. Parallel and Distributed AI: Psychological Modeling – Parallelism in Reasoning systems – Distributed Reasoning systems.

### UNIT V:

Expert System – Features of an Expert System – Persons Involved in Expert System Building Organizing Knowledge – Representing Knowledge – Frame based Methods – Basic characteristics of an Expert System – Expert Systems make mistakes – Knowledge Representation using Rules – Knowledge Representation using Semantic Nets – Knowledge Representation using Frames.

### Outcomes:

At the end of the course, the student should be able to:

- Gain a working knowledge of the foundations of, and modern applications in, artificial intelligence, including agent design, heuristic search, knowledge representation, planning, logic, natural language processing and machine learning.
- Gain experience writing several AI applications in a variety of programming languages.

### Text Books:

Artificial Intelligence, Elaine Rich and Kevin Knight, Tata McGraw Hill, 2<sup>nd</sup> Edition, 1991.  
A Guide to Expert Systems, Donald A. Waterman, Pearson Education.

# NEURAL NETWORK AND FUZZY SYSTEM

## OBJECTIVES:

To gain mastery over various fundamental concepts of fuzzy logic and thereto artificial neural network systems. This will help to get sufficient knowledge to analyze and design various intelligent control systems

## UNIT I FUNDAMENTALS OF FUZZY LOGIC

Basic concepts: fuzzy set theory- basic concept of crisp sets and fuzzy sets- complements- union- intersection- combination of operation- general aggregation operations- fuzzy relations-compatibility relations-orderings- morphisms- fuzzy relational equations-fuzzy set and systems

## UNIT II ARCHITECTURE OF NEURAL NETWORKS

Architectures: motivation for the development of natural networks-artificial neural networks-biological neural networks-area of applications-typical Architecture-setting weights-common activations functions- Basic learning rules- Mcculloch-Pitts neuron- Architecture, algorithm, applications-single layer net for pattern classification- Biases and thresholds, linear separability - Hebb's rule- algorithm -perceptron - Convergence theorem-Delta rule

## UNIT III BASIC NEURAL NETWORK TECHNIQUES

Back propagation neural net-standard back propagation-architecture algorithm- derivation of learning rules- number of hidden layers--associative and other neural networks- hetro associative memory neural net, auto associative net- Bidirectional associative memory- applications-Hopfield nets-Boltzman machine

## UNIT IV COMPETITIVE NEURAL NETWORKS

Neural network based on competition: fixed weight competitive nets- Kohonenself organizing maps and applications-learning vector quantization-counter propagation nets and applications adaptive resonance theory: basic architecture and operation-architecture, algorithm, application and analysis of ART1 & ART2

## **UNIT V SPECIAL NEURAL NETWORKS**

Cognitron and Neocognitron - Architecture, training algorithm and application- fuzzy associate memories, fuzzy system architecture- comparison of fuzzy and neural systems.

### **OUTCOMES:**

1. To understand the basic concept of fuzzy sets, fuzzy logic & defuzzification
2. To learn basics of Artificial Neural of theory and programming of Microprocessors
3. To analyze various techniques in feedback and feed forward Neural networks.
4. To understand the principle of competitive neural networks and Adaptive resonance theory
5. To learn the architecture and algorithm of Cognitron, Neo cognitron The concepts of fuzzy associative memory and fuzzy systems.

Text book(s) and/or required material

1. T. Klirvan- Fuzzy System & Fuzzy logic Prentice Hall of India, First Edition.
2. Lawrence Fussett- Fundamental of Neural networks, Prentice Hall , First Edition.

Reference Books:

- Bart Kosko, -Neural network and Fuzzy System - Prentice Hall-1994.
- J.Klin and T.A.Folger, -Fuzzy sets University and information- Prentice Hall -1996.
- J.M.Zurada, -Introduction to artificial neural systems -Jaico Publication house,Delhi 1994.
- VallusuRao and HayagvnaRao , -C++ Neural network and fuzzy logic -BPB and Publication, New Delhi,1996.

## **.NET FRAMEWORK LAB**

### **OBJECTIVES:**

This Lab course will help students to achieve the following objectives:

1. To introduce .Net IDE Component Framework.
2. Programming concepts of .Net Framework.
3. Creating website using ASP.Net Controls.

### **LIST:-**

1. Create a windows form with the following controls Textbox, Radio button, Check box, Command Button
2. Write a program for Menu option.
3. Create a program to connect with database and manipulate the records in the database using ADO .NET
4. Create a program to implement the concepts of OOPS for creating class, inheritance
5. Create a program to perform input validation using procedure.
6. Write a program to open a file and using I/O operations write contents into a file and read the contents from the file.
7. Create a window form using HTML controls.
8. Create a program to perform validation using validation controls.
9. Create a program in ASP .NET to connect with the database using ADODB connectivity and manipulate the records.
10. Write a program to store the employee details using class and methods in C# .NET
11. Write a program to Handle Exceptions
12. Write a program to create a form with Basic controls. In c#. NET.

### **OUTCOMES:**

At the end of this Lab course students will be able to:

2. Create user interactive web pages using ASP.Net.
3. Create simple data binding applications using ADO.Net connectivity.
4. Performing Database operations for Windows Form and web applications.