Osmania University

Faculty of Informatics

Two years MCA Program

Master of Computer Applications

Syllabi for Semesters – III and IV

With Effect from Academic Year 2023 – 2024

Osmania University
Hyderabad

SCHEME OF INSTRUCTION
MASTER OF COMPUTER APPLICATIONS (MCA)
SEMESTER- III

SNo	Course Code	Course Title	Hours/ Week			Scheme of Examination			No of Credits
						Max Marks		Duration (hrs)	
THEORY			**L**	**T**	**P**	**CIE**	**SEE**	**SEE**	**Cr**
1	PCC301	Software Engineering	**4**	**-**	**-**	**30**	**70**	**3**	**4**
2	PCC302	Computer Networks	**4**	**-**	**-**	**30**	**70**	**3**	**4**
3	PCC303	Artificial Intelligence	**4**	**-**	**-**	**30**	**70**	**3**	**4**
4	PCC304	Web Technologies	**4**	**-**	**-**	**30**	**70**	**3**	**4**
5	**PEC****	**Professional Elective–I**	**3**	**-**	**-**	**30**	**70**	**3**	**3**
6	**PEC****	**Professional Elective–II**	**3**	**-**	**-**	**30**	**70**	**3**	**3**
PRACTICALS									
7	LCC351	Computer Networks Lab	**-**	**-**	**3**	**25**	**50**	**3**	**1.5**
8	LCC352	Software Engineering Lab	**-**	**-**	**3**	**25**	**50**	**3**	**1.5**
9	LCC353	Web Technologies Lab	**-**	**-**	**3**	**25**	**50**	**3**	**1.5**
10	**SIP321**	Summer Internship	**-**	**-**	**-**	**50**		**-**	**1.5**
			20	**-**	**9**	**305**	**570**	**-**	**28**

Professional Electives	**Course Code-PEC****	**Professional Elective -1**	
	PEC311	Software Quality & Testing	
	PEC312	Distributed Systems	
	PEC313	Internet of Things	
	PEC314	Image Processing	
Professional Electives	**Course Code-PEC****	**Professional Elective – II**	
	PEC321	Network Security	
	PEC322	Cyber Security	
	PEC323	Information Retrieval System	
	PEC324	Natural Language Processing	

Abbreviation	**Full Form**	**Abbreviation**	**Full Form**
PCC	Professional Core Course	CIE	Continuous Internal Evaluation
PEC	Professional Elective Course	SEE	Semester End Evaluation
MGC	Management Course	L	Lecture
LCC	Laboratory Core Course	P	Practical

With effect from the academic year 2023-2024

SCHEME OF INSTRUCTION

MASTER OF COMPUTER APPLICATIONS (MCA)
SEMESTER- IV

SNo	Course Code	Course Title	Hours/ Week		Scheme of Examination			No of Credits
					Max Marks		Duration (hrs)	
THEORY			**L**	**P**	**CIE**	**SEE**	**SEE**	**Cr**
1	**PEC****	**Professional Elective –III**	**3**	**-**	**30**	**70**	**3**	**3**
2	**PEC****	**Professional Elective –IV**	**3**	**-**	**30**	**70**	**3**	**3**
3	**OE****	**Open Elective**	**2**	**-**	**30**	**70**	**3**	**2**
PRACTICALS								
4	Proj401	Project Work	**-**	**24**	**50**	**100**	**3**	**12**
		Total	**8**	**24**	**140**	**310**	**-**	**20**

Professional Electives

Course Code-PEC**	**Professional Elective – III**
PEC411	Block Chain Technologies
PEC412	Big Data Analytics
PEC413	Cloud Computing
PEC413	Deep Learning

Course Code-PEC**	**Professional Elective – IV**
PEC421	Distributed Database Systems
PEC422	Digital Forensics
PEC423	Optimization Techniques
PEC424	Enterprise Architecture

Course Code-OE**	**Open Elective**
OE 431	Professional Ethics
OE 432	Constitution of India
OE 433	Disaster Management
OE 434	Organization Behaviour
OE 435	Intellectual Property & Cyber Law
OE 436	Environmental Science

With effect from academic year 2023-2024

PCC301 **Software Engineering**

Credits : 4

Instruction4L hrs per week Duration of SEE3 hours

CIE30 marks SEE70 marks

Course Objectives

1. Learn the software problem and addressing it through various software processes
2. Study the SRS and software architecture
3. Understand planning and designing a software project
4. Comprehend the testing strategies and the need for performing testing
5. Learn how to carry out reengineering to the system and maintain it

Course Outcomes – Students will learn to

1. Apply software processes to solve software problem
2. Create SRS document and software architecture
3. Perform software planning in terms of staffing and scheduling
4. Create test cases and procedures
5. Re-engineer the developed software

Unit I

The software Problem: Cost, Schedule and Quality, Scale and change,

Software Processes: Process and project, Component Software Processes, SoftwareDevelopment Process Models, Project management Process.

Unit II

Software Requirements Analysis and Specification: Value of a good SRS, RequirementsProcess, Requirements Specification, Functional Specification with Use Cases, Other approaches for analysis.

Software Architecture: Role of Software Architecture Views, Component and connector view,Architectural styles for C & C view, Documenting Architecture Design, Evaluating Architectures.

Unit III

Planning a Software Project: Effort Estimation, Project Schedule and staffing, Quality Planning,Risk Management Planning, Project Monitoring Plan, Detailed Scheduling. **Design**: Design concepts, Function oriented Design, Object Oriented Design, Detailed Design, Verification, Metrics.

Unit IV

Coding and Unit Testing: Programming Principles and Guidelines, incrementally developing code, managing evolving code, unit testing, code inspection, Metrics. **Testing**: Testing Concepts, Testing Process, Black Box testing, White box testing, Metrics.

Unit V

Maintenance and Re-engineering: Software Maintenance, supportability, Reengineering,Business process Reengineering, Software reengineering, Reverse engineering; Restructuring, Forward engineering, Economics of Reengineering.

Software Process Improvement: Introduction, SPI process, CMMI, PCMM, Other SPIFrameworks, SPI return on investment, SPI Trends.

Suggested Reading

1. Pankaj Jalote, "Software Engineering- A Precise Approach", Wiley India, 2010.
2. Roger. S.Pressman , "Software Engineering - A Practitioner's Approach", 7th Edition, McGraw Hill Higher Education, 2010.
3. Deepak Jain, "Software Engineering", Oxford University Press, 2008.
4. Rajib Mall, "Fundamentals of Software Engineering", 4th Edition, PHI Learning, 2014.
5. Ian Sommerville, "Software Engineering", 10th Edition, Addison Wesley, 2015.

PCC302 **Computer Networks**

Credits : 4

Instruction	4L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Comprehend the fundamentals of computer networks
2. Learn the aspects relevant to physical and datalink layer
3. Understand network layer and its significance and functionality
4. Study transport layer and its operations
5. Learn the protocols implemented at application layer

Course Outcomes - Upon completion of the course, students will be able to:

1. Elaborate the network model
2. Explain transmission media and functions of datalink layer
3. Create routing tables based on DVR and LSR
4. Describe TCP and UDP protocols
5. Explain application layer protocols

Unit I

Data Communications: Components - Direction of Data flow - networks - Components and Categories - types of connections - Topologies -Protocols and Standards - ISO/OSI model, TCP/IP. **Transmission Media** - Coaxial Cable - Fiber Optics - Line Coding - Modems - RS232 Interfacing.

Unit II

Datalink Layer: Error detection and correction, CRC, Hamming code,

Flow Control and Error control , Stop and Wait protocol, Sliding Window protocol -go back-N ARQ - selective repeat ARQ .

MAC Layer: LAN - Pure and Slotted ALOHA, Ethernet IEEE 802.3 LAN Ethernet Efficiency Calculation,Bridges. ARP, RARP

Unit III

Network Layer: - Distance Vector Routing, Link State Routing,

IP v4 addressing, Subnetting, CIDR., Introduction to IPv6

ICMP , IGMP,OSPF and BGP.

Unit IV

Transport Layer: Services of transport layer, Multiplexing. Transmission Control Protocol (TCP) Congestion Control, timer management, Quality of services (QOS) and User Datagram Protocol (UDP)

Unit V

Socket Programming: Primitive and Advance System calls, Iterative and concurrent client server programs

Application Layer: Domain Name Space (DNS) - SMTP - FTP - HTTP

Suggested Readings

1. Andrew S. Tanenbaum, "Computer Networks", Pearson Education; Fourth Edition, 2008.
2. Behrouz A. Forouzan, "Data communication and Networking", Tata McGraw-Hill, 2009.

3. James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Pearson Education, 2006.
4. W Richard Stevents, Unix Network Programming,PHI,2003

PCC303 **Artificial Intelligence**

Credits : 4

Instruction	4L hrs per week	Instruction	4L hrs per week
CIE	30 marks	CIE	30 marks

Course Objectives:

- To familiarize the principles of Artificial Intelligence
- To study the techniques for knowledge representation and inference
- To learn the techniques involved in the creation of intelligent systems
- To study different applications like Game Playing Expert Systems, machine learning and natural language processing

Course outcomes :

Student will be able to

- Identify problems that are amenable to solution by AI method
- Understand and analyze working of an AI technique
- Formalize a given problem in the language/framework of different AI methods

UNIT-I

Introduction, History, Intelligent Systems, Foundations of AI, Sub-areas of AI, Applications, 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. Game Playing, Bounded Look-ahead Strategy and use of Evaluation Functions, Alpha-Beta Pruning

UNIT – II

Logic Concepts and Logic Programming: Introduction, Propositional Calculus, Propositional Logic, Natural Deduction System, Axiomatic System, Semantic Tableau System in Propositional Logic, Resolution Refutation in Propositional Logic, Predicate Logic, Logic Programming.

Knowledge Representation: Introduction, Approaches to Knowledge Representation, Knowledge Representation using Semantic Network, Knowledge Representation using Frames

UNIT – III

Expert System and Applications: Introduction, Phases in Building Expert Systems, Expert System Architecture, Expert Systems vs Traditional Systems, Truth Maintenance Systems, Application of Expert Systems, List of Shells and Tools. Uncertainty Measure-Probability Theory: Introduction, Probability Theory, Bayesian Belief Networks, Certainty Factor Theory, Dempster-Shafer Theory.

UNIT – IV

Machine-Learning Paradigms: Introduction, Machine Learning Systems, Supervised and Unsupervised Learning, Inductive Learning, Learning Decision Trees (Suggested Reading 2), Deductive Learning, Clustering, Support Vector Machines.

Artificial Neural Networks: Introduction, Artificial Neural Networks, Single-Layer Feed-Forward Networks, Multi-Layer Feed-Forward Networks, Radial-Basis Function Networks, Design Issues of Artificial Neural Networks, Recurrent Networks.

UNIT – V

Advanced Knowledge Representation Techniques: Case Grammars, Semantic Web.

Natural Language Processing: Introduction, Sentence Analysis Phases, Grammars and Parsers, Types of Parsers, Semantic Analysis, Universal Networking Knowledge.

Suggested Readings:

1. Saroj Kaushik, Artificial Intelligence, Cengage Learning, 2011
2. Russell, Norvig, Artificial Intelligence- A Modern Approach, Pearson Education, 2nd Edition,2004
3. Rich, Knight, Nair, Artificial Intelligence, Tata McGraw Hill, 3rd Edition, 2009

With effect from academic year 2023-2024

PCC304 **Web Technologies**

Credits : 3

Instruction	4(3L+1T) hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Learn basics of HTML and DHTML
2. Understand the workings of event model
3. Study the java scripting language
4. Learn the VB scripts
5. Comprehend the active server pages

Course Outcomes

1. Write HTML and DHTML programs
2. Create programs on event models
3. Implement java script programs
4. Write VB script programs
5. Create ASP programs

Unit I

HTML: Markup languages, common tags, header, test styling, linking images Formatting text, Unordered lists, nested and ordered list, Tabs-and formatting, Basic forms; Complex forms linking, Meta Tags. **Dynamic HTML**: Cascading style sheets in line styles, style element, External Style sheet, text flow and Box model, user style sheets.

Unit II

Object model and collections: Object referencing, collections all, children frames, navigator object. **Event model**: ONCLICK, ONLOAD, Error Handling, ON ERRORS ONMOUSEMOVE, ONMOUSEOVER, ONMOUSEOUT, ONFOCUS, ONBLUR, ONSUBMIT. **Dynamic HTML**: Filters and transitions, Data binding with Tabular data control binding to IMO, TABLE, Structured graphics, Active controls.

Unit III

Introduction to scripting, Java Script, Data types, Arithmetic's Equality relational, assignment increment, decrement operators, Java Script Control Structures- if, if-else, while. Java Script **Control Structures**: For, Switch, Do/while, break.

Programming modules, recursion, recursion vs iteration global functions arrays, using arrays, Reference and reference parameters, passing arrays to functions, multiple subscripted arrays, objects-math, string. Boolean and number.

Unit IV

Client side scripting with VB Script, operations, Data types and control structures, Functions, Arrays, String manipulations, classes and objects. **Web Servers**: Personal Web server, Internet information server, Apache Web Server, Installation of a Web Server.

Unit V

Active Sever Pages, Client side Scripting vs Server side Scripting, Server side Active X Component, ADO, file system objects, Session tracking, CGI and PERL5, String Processing and Regular Expressions, Server side includes, Cookies and PERL XML Document Type Definition, XML Parsers, Using XML with HTML.

Suggested Readings

1 Deiterl, Deitel& NIETO, "Internet & World Wide Web - How to Program", Pearson Education, Third Edition, 2004.

2 Steven Holzner, "HTML Black Book - Comprehensive Problem Server", Dream Tech Press, 2000.

3 B Sosinsky, V Hilley, "Programming the Web - An Introduction", MGH, 2004.

PEC311 **Software Quality and Testing**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Learn the essentials of software quality
2. Study methods to integrate software quality activities in the project
3. Understand the software quality metrics
4. Learn building software testing strategy
5. Comprehend testing various artifacts of a software project

Course Outcomes

1. Explain the essentials of software quality
2. Elaborate the methods to integrate software quality activities in the project
3. Describe the software quality metrics
4. Discuss building software testing strategy
5. Perform testing various artifacts of a software project

UNIT - I

The Software Quality Challenge, Introduction Software Quality Factors, The Components of the Software Quality Assurance System – Overview, Development and Quality Plans.

UNIT - II

Integrating Quality Activities in the Project Life Cycle, Assuring the Quality of Software Maintenance Components, CASE Tools and their effect on Software Quality, Procedure and Work Instructions, Supporting Quality Devices, Configuration Management, Documentation Control, Project Progress Control.

UNIT - III

Software Quality Metrics, Costs of Software Quality, Quality Management Standards - ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma, SQA Project Process Standards – IEEE Software Engineering Standards.

UNIT - IV

Building a Software Testing Strategy, Establishing a Software Testing Methodology, Determining Your Software Testing Techniques, Eleven – Step Software Testing Process Overview, Assess Project Management Development Estimate and Status, Develop Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report Test Results, Test Software Changes, Evaluate Test Effectiveness.

UNIT - V

Testing Client / Server Systems, Testing the Adequacy of System Documentation, Testing Web-based Systems, Testing Off – the – Shelf Software, Testing in a Multiplatform Environment, Testing Security, Testing a Data Warehouse, Creating Test Documentation, Software Testing Tools, Taxonomy of Testing Tools, Methodology to Evaluate Automated Testing Tools, Load Runner, Win Runner and Rational Testing Tools, Java Testing Tools, JMetra, JUNIT and Cactus.

Suggested Reading

1. Daniel Galin, Software Quality Assurance – From Theory to Implementation, Pearson Education.2004
2. Mordechai Ben – Menachem / Garry S.Marliss, Software Quality – Producing Practical, Consistent Software, BS Publications, 2014
3. William E. Perry, Effective Methods for Software Testing, 3 rd Edition, 2006, Wiley .
4. Srinivasan Desikan, Gopalaswamy Ramesh, Software Testing, Principles and Practices, 2006. Pearson Education.
5. Dr.K.V.K.K. Prasad, Software Testing Tool, Wiley Publishers

PEC312 **Distributed Systems**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

CourseObjectives

1. Understand the architecture, processes and communication of distributed system
2. Learn the naming and synchronization strategies
3. Study fault tolerance, and distributed object based system
4. Learn distributed file system and distributed web based system
5. Comprehend the distributed coordination based system and map reduce

CourseOutcomes

1. Explain the architecture, processes and communication of distributed system
2. Elaborate the naming and synchronization strategies
3. Describe the fault tolerance and distributed object based system
4. Discuss the distributed file system and distributed web based system
5. Explain distributed coordination based system and map reduce

Unit I

Introduction: Goals and Types of Distributed Systems

Architectures: Architectural Styles, System Architectures, Architectures versus Middleware, and Self-Management in Distributed Systems.

Processes: Threads, Virtualization, Clients, Servers, and Code Migration.

Communication: Fundamentals, Remote Procedure Call, Message-Oriented Communication, Stream-Oriented Communication, and Multicast Communication.

Unit II

Naming: Names, Identifiers and Addresses, Flat Naming, Structured Naming, and Attribute-Based Naming.

Synchronization: Clock Synchronization, Logical Clocks, Mutual Exclusion, Global Positioning of Nodes, and Election Algorithms. **Consistency and Replication:** Introduction, Data-Centric Consistency Models, Client-Centric Consistency Models, Replica Management, and Consistency Protocols.

Unit III

Fault Tolerance: Introduction to Fault Tolerance, Process Resilience, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit, and Recovery.

Distributed Object-Based Systems: Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance, and Security.

Unit IV

Distributed File Systems: Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance, and Security.

Distributed Web-Based Systems: Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance, and Security.

Unit V

Distributed Coordination-Based Systems: Introduction to Coordination Models, Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance, and Security.

Map-Reduce: Example, Scaling, programming model, Apache Hadoop, Amazon Elastic Map Reduce, Mapreduce.net, Pig and Hive.

Suggested Readings

1. Andrew S. Tanenbaum and Maarten Van Steen, ―Distributed Systems‖, PHI 2nd Edition, 2009.
2. R.Hill, L.Hirsch, P.Lake, S.Moshiri, ―Guide to Cloud Computing, Principles and Practice‖, Springer, 2013.
3. R.Buyya, J.Borberg, A.Goscinski,‖Cloud Computing-Principles and Paradigms‖,Wiley 2013.

PEC313 **Internet of Things**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Discuss fundamentals of IoT and its applications and requisite infrastructure
2. Describe Internet principles and communication technologies relevant to IoT
3. Discuss hardware and software aspects of designing an IoT system
4. Describe concepts of cloud computing and Data Analytics
5. Discuss business models and manufacturing strategies of IoT products

Course Outcomes

Student will be able to

1. Understand the various applications of IoT and other enabling technologies.
2. Comprehend various protocols and communication technologies used in IoT
3. Design simple IoT systems with requisite hardware and C programming software
4. Understand the relevance of cloud computing and data analytics to IoT
5. Comprehend the business model of IoT from developing a prototype to launching a product.

UNIT- I

Introduction to Internet of Things

IOT vision, Strategic research and innnovation directions, Iot Applications, Related future technologies, Infrastructure, Networks and communications, Processes, Data Management, Security, Device level energy issues.

UNIT- II

Internet Principles and communication technology

Internet Communications: An Overview – IP,TCP,IP protocol Suite, UDP. IP addresses – DNS, Static and Dynamic IP addresses, MAC Addressess, TCP and UDP Ports, Application Layer Protocols HTTP,HTTPS, Cost Vs Ease of Production, Prototypes and Production, Open Source Vs Closed Source.

UNIT- III

Prototyping and programming for IoT

Prototyping Embedded Devices – Sensors, Actuators, Microcontrollers, SoC, Choosing a platform, Prototyping, Hardware platforms – Arduino, Raspberry Pi. Prototyping the physical design – Laser Cutting, 3D printing, CNC Milling.

Techniques for writing embedded C code: Integer data types in C, Manipulating bits - AND,OR,XOR,NOT, Reading and writing from I/ O ports. Simple Embedded C programs for LED Blinking, Control of motor using switch and temperature sensor for arduino board.

UNIT- IV

Cloud computing and Data analytics

Introduction to Cloud storage models -SAAS, PAAS, IAAS. Communication APIs, Amazon webservices for IoT, Skynet IoT Messaging Platform.

Introduction to Data Analytics for IoT - Apache hadoop- Map reduce job execution workflow.

UNIT- V

IoT Product Manufacturing - From prototype to reality

Business model for IoT product manufacturing, Business models canvas, Funding an IoT Startup, Mass manufacturing - designing kits, designing PCB,3D printing, certification, Scaling up software, Ethical issues in IoT- Privacy, Control, Environment, solutions to ethical issues.

Suggested Readings

1. Internet of Things - Converging Technologies for smart environments and Integrated ecosystems, River Publishers.
2. Designing the Internet of Things , Adrian McEwen, Hakim Cassimally. Wiley India Publishers
3. Fundamentals of embedded software: where C meets assembly by Daneil W lewies, Pearson.
4. Internet of things -A hands on Approach, ArshdeepBahga, Universities press.

EC323 **Image Processing**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Understand image processing fundamentals
2. Understand image transforms
3. Understand image enhancement
4. Understand image restoration and feature extraction
5. Understand image reconstruction

Course Outcomes

1. Learn image processing fundamentals
2. Learn image transforms
3. Learn image enhancement
4. Learn image restoration and feature extraction
5. Learn image reconstruction

Unit I

Fundamentals- Need for DIP- Fundamental steps in DIP – Elements of visual perception -Image sensing and Acquisition – Image Sampling and Quantization – Imaging geometry, discrete image mathematical characterization.

Unit II

Image Transforms - Two dimensional Fourier Transform- Properties – Fast Fourier Transform – Inverse FFT,Discrete cosine transform and KL transform.-Discrete Short time Fourier Transform-Wavelet Transform- Discrete wavelet Transform- and its application in Compression.

Unit III

Image Enhancement - **Spatial Domain**: Basic relationship between pixels- Basic Gray level Transformations – Histogram Processing – Smoothing spatial filters- Sharpening spatial filters. **Frequency Domain**: Smoothing frequency domain filters- sharpening frequency domain filters Homomorphic filtering.

Unit IV

Image Restoration:- Overview of Degradation models –Unconstrained and constrained restorations-Inverse Filtering ,WienerFilter.

Feature Extraction: - Detection of discontinuities – Edge linking and Boundary detection-Thresholding- -Edge based segmentation-Region based Segmentation- matching-Advanced optimal border and surface detection- Use of motion in segmentation. Image Morphology – Boundary descriptors- Regional descriptors.

Unit V

Image Reconstruction from Projections: - Need- Radon Transform – Back projection operator-Projection Theorem- Inverse Radon Transform.

Suggested Reading

1. Rafael C.Gonzalez& Richard E.Woods – Digital Image Processing – Pearson Education- 2/e – 2004.
2. Anil.K.Jain – Fundamentals of Digital Image Processing- Pearson Education-2003.
3. B.Chanda&D.DuttaMajumder – Digital Image Processing and Analysis – Prentice Hall of India – 2002
4. William K. Pratt – Digital Image Processing – John Wiley & Sons-2/e, 2004

PEC321 **Network Security**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

CourseObjectives

1. Understand the significant aspects of network security
2. Comprehend secret and public key cryptography
3. Learn hash functions and digital signatures
4. Study the digital signatures and smart cards
5. Comprehend the applications of network applications

CourseOutcomes

1. Explain the fundamentals of network security
2. Elaborate the concepts secret and public key cryptography
3. Elucidate the hash functions digital signatures
4. Describe the digital signatures and smart cards
5. Explain the applications of network security

UNIT-I

Introduction: Attributes of Security, Integrity, Authenticity, Non-repudiation, Confidentiality Authorization, Anonymity, Types of Attacks, DoS, IP Spoofing, Replay, Man-in-the-Middle attacks General Threats to Computer Network, Worms, Viruses, -Trojans

UNIT-II

Secret Key Cryptography :DES, Triple DES, AES, Key distribution, Attacks

Public Key Cryptography: RSA, ECC, Key Exchange (Diffie-Hellman), Java Cryptography Extensions, Attacks

UNIT-III

Integrity, Authentication and Non-Repudiation : Hash Function (MD5, SHA5), Message Authentication Code (MAC), Digital Signature (RSA, DSA Signatures), Biometric Authentication.

UNIT-IV

PKI Interface: Digital Certificates, Certifying Authorities, POP Key Interface, System Security using Firewalls and VPN's.

Smart Cards: Application Security using Smart Cards, Zero Knowledge Protocols and their use in Smart Cards, Attacks on Smart Cards

UNIT-V

Applications: Kerberos, Web Security Protocols (SSL), IPSec, Electronic Payments, E-cash, Secure Electronic Transaction (SET), Micro Payments, Case Studies of Enterprise Security (.NET and J2EE)

Suggested Reading

1. William Stallings, Cryptography and Network Security, 4th Edition. Pearson,. 2009.
2. Behrouz A Forouzan, Cryptography and Network Security, TMH, 2009
3. Joseph MiggaKizza, A Guide to Computer Network Security, Springer, 2010
4. Dario Cataiano, Contemporary Cryptology, Springer, 2010.

PEC421 **Cyber Security**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Understand the policies and security evolution
2. Learn cyber security objectives and guidance
3. Study policy catalog and issues
4. Comprehend cyber management and infrastructure issues
5. Learn the cyber security case studies

Course Outcomes

1. Explain the policies and security evolution
2. Describe cyber security objectives and guidance
3. Discuss policy catalog and issues
4. Elaborate cyber management and infrastructure issues
5. Elucidate the case studies on cyber security

Unit I: Policies and Security Evolution

Introduction - Cyber Security, Cyber Security policy, Domain of Cyber Security Policy, Laws and Regulations

Cyber Security Evolution - Enterprise Policy, Technology Operations, Technology Configuration, Strategy Versus, Policy, Cyber Security Evolution, Productivity, Internet, E-Commerce, Counter Measures, Challenges.

Unit II: Cyber Security Objectives and Guidance

Security Objectives - Cyber Security Metrics, Security Management Goals, Counting Vulnerabilities, Security Frameworks, E-Commerce Systems, Industrial Control Systems, Personal Mobile Devices, Security Policy Objectives, Guidance for Decision Makers, Tone at the Top, Policy as a Project.

Catalog Approach - Cyber Security Management, Arriving at Goals, Cyber Security Documentation, the Catalog Approach, Catalog Format, Cyber Security Policy Taxonomy

Unit III: Policy Catalog and Issues

Cyber Security Policy Catalog - Cyber Governance Issues, Net Neutrality, Internet Names and Numbers, Copyright and Trademarks, Email and Messaging, Cyber User Issues, Malvertising, Impersonation.

Cyber user and conflict Issues - Appropriate Use, Cyber Crime, Geo location, Privacy, Cyber Conflict Issues, Intellectual property Theft, Cyber Espionage, Cyber Sabotage, Cyber Welfare.

Unit IV: Cyber Management and Infrastructures Issues

Cyber Management Issues - Fiduciary Responsibility – Risk Management – Professional Certification – Supply Chain – Security

Cyber Infrastructure Issues - Principles – Research and Development – Cyber Infrastructure Issue – Banking and finance – Health care – Industrial Control systems.

Unit V: Case Study

Government’s Approach to Cyber Security Policy - Cyber security strategy-Brief history-Public policy development in the U.S Federal Government.

Espionage - The rise of cyber crime- Espionage and Nation-state Actions-Policy response to growing Espionage threats-Congressional Action.

Suggested Readings

1. Jennifer L. Bayuk, J. Healey, P. Rohmeyer, Marcus Sachs, Jeffrey Schmidt, Joseph Weiss "Cyber Security Policy Guidebook" John Wiley & Sons 2012.
2. Rick Howard "Cyber Security Essentials" Auerbach Publications 2011.
3. Richard A. Clarke, Robert Knake"Cyberwar: The Next Threat to National Security & What to Do About It" Ecco 2010
   Dan Shoemaker "Cyber security The Essential Body of Knowledge", 1st edition, Cengage Learning 2011.

PEC314 **Information Retrieval System**

Credits : 3

Instruction	3Lhrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Understand IR strategies
2. Study basic retrieval utilities
3. Learn cross language IR
4. Comprehend efficiency aspects
5. Learn distributed IR

Course Outcomes

1. Explain IR strategies
2. Elucidate basic retrieval utilities
3. Discuss cross language IR
4. Describe efficiency aspects
5. Elaborate distributed IR

UNIT-I

Introduction to Retrieval. Strategies: Vector Space model, Probabilistic Retrieval.
Strategies Language Models: Simple Term Weights, Non Binary Independence Model.

UNIT-II

Retrieval Utilities: Relevance Feedback, Clustering, N-grams, Regression Analysis, Thesauri.

UNIT-III

Retrieval Utilities: Semantic Networks, Parsing, Cross-Language Information Retrieval: Introduction, Crossing the Language Barrier.

UNIT-IV

Efficiency: Inverted Index, Query Processing, Signature Files, Duplicate Document Detection.

UNIT - V

Integrating Structured Data and Text: A Historical Progression, Information Retrieval as a Relational Application, Semi-Structured Search using a Relational Schema.
Distributed Information Retrieval: A Theoretical Model of Distributed Retrieval, Web Search.

Suggested Reading:

1. David A. Grossman, Ophir Frieder. “Information Retrieval - Algorithms and Heuristics”, Springer, 2nd Edition (Distributed by Universities Press), 2004.
2. Gerald J Kowalski, Mark T Maybury. “Information Storage and Retrieval Systems”, Springer, 2000.
3. SoumenChakrabarti, “Mining the Web: Discovering Knowledge. from Hypertext Data", Morgan-Kaufmann Publishers, 2002.
4. Christopher D. Manning, PrabhakarRaghavan, HinrichSchGtze, “An Introduction to Information Retrieval”, Cambridge University Press, Cambridge, England,-2009.

PEC324 **Natural Language Processing**

Credits : 3

Instruction	3L hrs per week	Duration of SEE	3 hours
CIE	30 marks	SEE	70 marks

Course Objectives

1. Learn elementary probability and information theory
2. Study the linguistic essentials
3. Comprehend statistical inference and word sense disambiguation
4. Understand evaluation measures and markov models
5. Learn probabilistic context free grammars

Course Outcomes – Learners on completion of the course, be able to

1. Explain elementary probability and information theory
2. Discuss the linguistic essentials
3. Describe statistical inference and word sense disambiguation
4. Elaborate evaluation measures and markov models
5. Elucidate probabilistic context free grammars

UNIT I

Introduction of Elementary Probability Theory, Essential Information Theory. Linguistic Essentials Corpus-Based Work Collocations.

UNIT II

Statistical Inference: Bins: Forming Equivalence Classes, Reliability vs. Discrimination, n-gram models, Building ngram models, An Information Theoretic Approach.

UNIT III

Word Sense Disambiguation: Methodological Preliminaries, Supervised and unsupervised learning, Pseudo words, Upper and lower bounds on performance, Supervised Disambiguation, Bayesian classification.

UNIT IV

Evaluation Measures, Markov Models: Hidden Markov Models, Use, General form of an HMM Part-of-Speech Tagging

UNIT-V

Probabilistic Context Free Grammars: Introduction of Clustering **Information Retrieval:** Background, The Vector Space Model.

Suggested Reading

1. Christopher D. Manning, HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
2. James Allan, Natural Language Understanding, Pearson Education, 1994.
3. Tanveer Siddiqui, US Tiwary, Natural Language Processing and Information Retrieval, Oxford University Press, 2008.

LCC351 **Computer Networks Lab**

Credits : 2

Instruction	3P hrs per week	Duration of SEE	3 hours
CIE	25 marks	SEE	50 marks

Course Objectives

1. Understand basic commands of networks
2. Learn socket program implementation
3. Understand connection oriented socket programs
4. Learn connectionless socket programs
5. Understand DNS implementation

Course Outcomes - Upon completion of the course, the students will be able to:

1. Execute basic commands of networks
2. Implement socket program implementation
3. Execute connection oriented socket programs
4. Implement connection less socket programs
5. Execute DNS implementation

Programs to be written on the following concepts using any programming language like Python, C, C++, Java.

1. Understanding and using of commands like ifconfig, netstat, ping, arp, telnet, ftp, finger, traceroute, whois.
2. Socket Programming: Implementation of Connection-Oriented Service using standard ports.
3. Implementation of Connection-Less Service using standard ports.
4. Implementation of Connection-Oriented Iterative Echo-Server, date and time, character generation using user-defined ports.
5. Implementation of Connectionless Iterative Echo-server, date and time, character generation using user-defined ports.
6. Implementation of Connection-Oriented Concurrent Echo-server, date and time, character generation using user-defined ports.
7. Program for connection-oriented Iterative Service in which server reverses the string sent by the client and sends it back.
8. Program for connection-oriented Iterative service in which server changes the case of the strings sent by the client and sends back (Case Server).
9. Program for Connection-Oriented Iterative service in which server calculates the net-salary of an employee based on the following details sent by the client
   i) basic ii) hra iii) da iv) pt v) epf vi) net-salary=basic+hra+da-pt-epf).
10. Program for file access using sockets.
11. Program for Remote Command Execution using sockets .
12. Implementation of DNS.

LCC352 **Software Engineering Lab**

Credits : 2

Instruction	3P hrs per week	Duration of SEE	3 hours
CIE	25 marks	SEE	50 marks

Course Objectives

1. Learn use case diagram
2. Learn class and object diagram
3. Understand sequence and collaboration diagrams
4. Study state-chart and activity diagrams
5. Comprehend component and deployment diagrams

Course Outcomes

1. Apply use case diagram
2. Apply class and object diagram
3. Apply sequence and collaboration diagrams
4. Apply state-chart and activity diagrams
5. Apply component and deployment diagrams

1. Phases in software development project, overview, need, coverage of topics

2. To assign the requirement engineering tasks

3. To perform the system analysis: Requirement analysis, SRS

4. To perform the function-oriented diagram: DFD and Structured chart

5. To perform the user’s view analysis: Use case diagram

6. To draw the structural view diagram: Class diagram, object diagram

7. To draw the behavioral view diagram: Sequence diagram, Collaboration diagram

8. To draw the behavioral view diagram: State-chart diagram, Activity diagram

9. To draw the implementation view diagram: Component diagram

10. To draw the environmental view diagram: Deployment diagram

11. To perform various testing using the testing tool unit testing, integration testing

Draw UML diagrams for the following system

1. ATM application
2. Library management system
3. Railway reservation
4. E-Commerce System
5. Banking System

Perform the following tasks

Background: Software has made the world a global village today. The impact of software spans across almost all aspect of human life. All organizations, Institutions and companies are leveraging the potentials of software in automating the critical functions and eliminating manual interventions. Software is also a predominant area for trade and export especially for the countries like India. Domains like health care, Airlines, financial Services, Insurance, retails, Education, and many more have exploited software and still there a lot of the scope for software to create impact and add values in multiple dimensions.

Problem Description: In the context of this background, identify the areas (or application or systems) how software has been leveraged extensively in the following domains

1. Health Care 2. Airlines 3. Banking Insurance
4. Retail 5. Education

Background: In the early years of computers applications, the focus of the development and innovation were on hardware. Software was largely views as an afterthought. Computer programming was an art. Programmers did not follow any disciplined or formalized approaches. This way of doing things was adequate for a while, until the sophisticated of computer applications outgrow. Software soon took over and more functions which were done manually. A software houses begin to develop for widespread distribution. Software development projects produced thousands of source program statement. With the increase in the size and complexity of the software, following situation resulted is collectively termed as software crisis.

1. Time Slippage
2. Cost Slippage
3. Failure at customer Site
4. Intractable Error after delivery

Problem Description: In the context of this background, for each of the scenario mentioned below, identify the most appropriate problem related to software crisis and mention the same in the table provided.

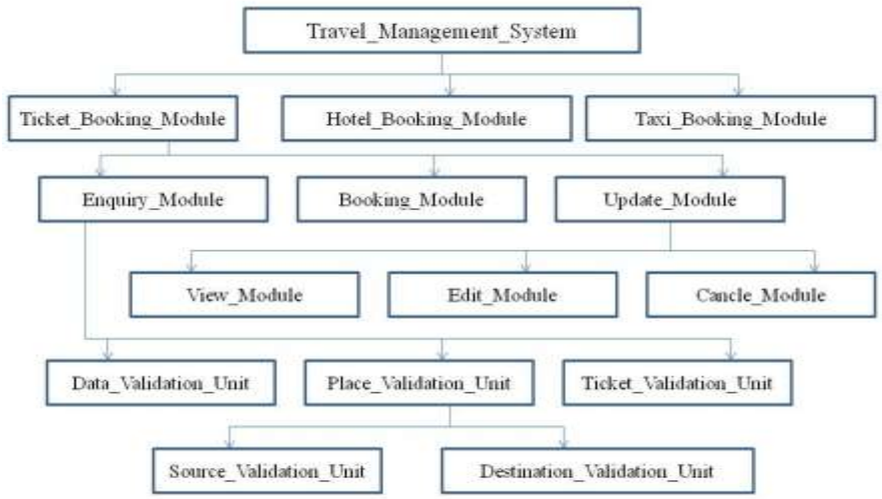
Scenario A: Railways reservation software was delivered to the customer and was installed in one of the metro station at 12.00 AM (mid-night) as per the plan. The system worked quite fine till the next day 12.00 PM (noon). The system crashed at 12.00 PM and the railways authorities could not continue using software for reservation till 02.00 PM. It took two hours to fix the defect in the software in the software.

Scenario B: A polar satellite launch vehicle was scheduled for the launch on August 15th. The auto-pilot of the rocket to be delivered for integration of the rocket on May 15th. The design and development of the software for the auto-pilot more effort because of which the auto-pilot was delivered for the integration on June 15th (delayed by a month). The rocket was launched on Sep 15th (delayed by a month).

Scenario C: Software for financial systems was delivered to the customer. Customer informed the development team about a mal-function in the system. As the software was huge and complex, the development team could not identify the defect in the software.

INTEGRATION TESTING

Background: Integration testing is carried out after the completion of unit testing and before the software is delivered for system testing. In top down integration testing, dummy stubs are required for bottom level modules. Similarly, in bottom up testing, dummy drivers are required for top level modules



Problem Description: Consider the scenario of development of software for Travel, Management System (TMS) is in progress. The TMS software has 3 major modules namely Ticket_Booking_Module, Hotel_Booking_Module and Taxi_Booking_Module. The Ticket_Booking_Module has 3 sub modules namely Enquiry_Module, Booking_Module and Update_Module. The enquiry module uses Date_Validation_Unit, Ticket_Validation_Unit and Place_Validation_Unit.

In the context of the given scenario, identify the usage of stub or driver for the following situations.

1. Except the Ticket_validation_Unit, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is in progress for the TMS software. To carry out the integration testing, which among the following is necessary?

2. The coding and unit testing of all the module, sub modules and units of TMS are completed except the Update_Module (coding and testing for Edit_Module, Cancel_Module and View_Module are also completed). The bottom-up integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing?

3. Except the Taxi_Booking_Module, the coding and unit testing of all other modules, sub modules and units of TMS are completed. The top-down integration is to be started for the TMS software. Mention any stub or driver needed to carry out the integration testing.

Background: Performance testing tests the non-functional requirements of the system. The different types of performance testing are load testing, stress testing, endurance testing and spike testing.

Problem Description: Identify the type of performance testing for the following:

1. A space craft is expected to function for nearly 8 years in space. The orbit control system of the spacecraft is a real-time embedded system. Before the launch, the embedded software is to be tested to ensure that it is capable of working for 8 years in the space. Identify the suitable performance testing category to be carried out to ensure that the space craft will be functioning for 8 years in the space as required.

2. Global Education Centre (GEC) at Infosys Mysore provides the training for fresh entrants. GEC uses an automated tool for conducting objective type test for the trainees. At a time, a maximum of 2000 trainees are expected to take the test. Before the tool is deployed, testing of the tool was carried out to ensure that it is capable of supporting 2000 simultaneous users. Indicate the performance testing category?

3. A university uses its web-based portal for publishing the results of the students. When the results of an examination were announced on the website recently on a pre-planned date, the web site crashed. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

4. During unexpected terrorist attack, one of the popular websites crashed as many people

logged into the web-site in a short span of time to know the consequences of terrorist attack and for immediate guidelines from the security personnel. After analyzing the situation, the maintenance team of that website came to know that it was the consequences of unexpected load on the system which had never happened previously. Which type of performance testing should have been done during web-site development to avoid this unpleasant situation?

Background: Enhancements are introduction of new features to the software and might be released in different versions. Whenever a version is released, regression testing should be done on the system to ensure that the existing features have not been disturbed.

Problem Description: Consider the scenario of development of software for Travel Management System (TMS) discussed in previous assignment. TMS has been developed by Infosys and released to its customer Advance Travel Solutions Ltd. (ATSL). Integration testing, system testing and acceptance testing were carried out before releasing the final build to the customer. However, as per the customer feedback during the first month of usage of the software, some minor changes are required in the Enquiry Module of the TMS. The customer has approached Infosys with the minor changes for upgrading the software. The development team of Infosys has incorporated. Those changes, and delivered the software to testing team to test the upgraded software. Which among the following statement is true?

a. Since minor changes are there, integration of the Enquiry Module and quick system testing on Enquiry module should be done.

b. The incorporation of minor changes would have introduced new bugs into other modules, so regression testing should be carried out.

c. Since the acceptance testing is already carried out, it is enough if the team performs sanity testing on the Enquire module.

d. No need of testing any module.

Background: There are some metrics which are fundamental and the rest can be derived from these. Examples of basic (fundamental) measures are size, effort, defect, and schedule. If the fundamental measures are known, then we can derive others. For example if size and effort are known, we can get Productivity (=size/effort). If the total numbers of defects are known we can get the Quality (=defect/size) and so on.

Problem Description: Online loan system has two modules for the two basic services, namely Car loan service and House loan service.

The two modules have been named as Car_Loan_Module and House_Loan_Module. Car_Loan_Module has 2000 lines of uncommented source code. House_Loan_Module has 3000 lines of uncommented source code. Car_Loan_Module was completely implemented by Mike. House_Loan_Module was completely implemented by John. Mike took 100 person hours to implement Car_Loan_Module. John took 200 person hours to implement House_Loan_Module. Mike‟s module had 5 defects. Jonh's module had 6 defects. With respect to the context given, which among the following is an INCORRECT statement?

Choose one:

1. John‟s quality is better than Mike.
2. John‟s productivity is more than Mike.
3. John introduced more defects than Mike.
4. John‟s effort is more than Mike.

With effect from academic year 2023-2024

LCC351 **Web Technologies Lab**

Credits : 2

Instruction	3P hrs per week	Duration of SEE	3 hours
CIE	25 marks	SEE	50 marks

1. Develop College Website using XHTML and CSS.
2. Develop HTML form with client validations using Java Script.
3. Publishing XML document using XSLT
4. XML document processing using SAX and DOM.
5. Text processing using Regular expressions and pattern matching.
6. Develop form processing application using CGI.pm
7. Develop CGI-Perl Web application with State and Session Tracking.
8. Develop a simple Java servlet application.
9. Develop Java servlet application with session tracking
10. Develop a simple JSP application.
11. Creation of an application access database with JDBC
12. Develop full fledged web application with database access spreading over to 3 sessions.

SIP321 **Summer Internship***

Credits : 2

Instruction	6-week
CIE	50 marks

Program Description

The Internship Program allows MCA students to gain practical experience in the workplace before receiving their graduate degrees. The internship is a required academic course. The student identifies companies willing to hire him/her on a full time basis for a 6-week period (minimum required), usually in the summer. The Internship Program supervises the students and awards academic credits (2) upon successful completion of all the required assignments.

Intended Learning Outcomes

Upon successful completion of the internship, you should be able to

1. Communicate a practical understanding of how a technology actually operates
2. Demonstrate the ability to integrate and apply theoretical knowledge and skills developed in various courses to real-world situations in a business organization
3. Exhibit the ability to effectively work in a professional environment and demonstrate work ethic and commitment in a work-based environment
4. Demonstrate the ability to successfully complete internship assignments.
5. Reflect on personal and professional development needs and set strategic goals for advancing along an intended career path
6. Communicate effectively in a professional environment in both English and regional language, orally and in writing.