Autonomous Program Structure
Third Year B. Tech. Fifth Semester
Computer Engineering
Academic Year: 2022-2023 Onwards

Course Code	Course Title	Teaching Scheme Hours /Week			Examination Scheme				Marks	Credits
		Lecture	Tutorial	Practical	In Semester	End Semester	Oral	Practical		
20CE501	Computer Networks	3	1	0	50	50	0	0	100	4
20CE502	Design and Analysis of Algorithms	3	0	0	50	50	0	0	100	3
20CE503	Software Design and Architecture	3	1	0	50	50	0	0	100	4
20PECE 501	Programme Elective-I	3	0	0	50	50	0	0	100	3
20PECE 502	Programme Elective-II*	3	0	0	50	50	0	0	100	3
20OEHS 501	Open HS Elective –I	3	0	0	50	50	0	0	100	3
20CE501L	Computer Networks Laboratory	0	0	2	25	0	0	25	50	1
20CE504L	Programming Skills Development- II Laboratory	0	0	4	25	0	25	0	50	2
20PECE 501L	Programme Elective Laboratory-I	0	0	2	25	0	25	0	50	1
	Total	**18**	**2**	**8**	**375**	**300**	**50**	**25**	**750**	**24**
	Grand Total	**28**			**750**					**24**

Programme Elective-I	Programme Elective-I Laboratory	Programme Elective-II
20PECE501A Digital Image Processing 20PECE501B Java Full Stack Technologies 20PECE501C Statistics for Computer Science 20PECE501D Linux Internals	20PECE501LA Digital Image Processing 20PECE501LB Java Full Stack Technologies 20PECE501LC Statistics for Computer Science 20PECE501LD Linux Internals	* NPTEL / Swayam Course

APPROVED BY
Secretary Governing Body
MKSSS's Cummins College of Engineering
For Women, Pune-411052

APPROVED BY
Chairman Governing Body
MKSSS's Cummins College of Engineering
For Women, Pune-411052

M.K.S.S.S. Cummins college of Engg. for Women
KARVENAGAR
Pune -52

Open Elective I (Humanities)

Sr. No.	Course Code	Course Title
1	20OEHS501A	Entrepreneurship Development
2	20OEHS501B	Intellectual Property Rights
3	20OEHS501C	Introduction to Digital Marketing
4	20OEHS501D	Law for Engineers
5	20OEHS501E	Organizational Behaviour
6	20OEHS501F	Project Management

APPROVED BY
Secretary Governing Body
MKSSS's Cummins College of Engineering
For Women, Pune-411052

APPROVED BY
Chairman Governing Body
MKSSS's Cummins College of Engineering
For Women, Pune-411052

M.K.S.S.S. Cummins college of Engg. for Women
KARVENAGAR
Pune -52

20CE 501 Computer Networks

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hours / Week	In Semester: 50 Marks
Tutorial: 1 Hours / Week	End Semester: 50 Marks
	Credits: 4

Course Objectives:

To facilitate the learner to

1. Distinguish the fundamental concepts of networking standards, protocols and technologies
2. Identify role of protocols at various layers in the protocol stack
3. Select and Compare the appropriate network and protocols by understanding the given requirements for a given system
4. Get familiar with fundamental concepts of network security and recent trends in networking

Course Outcomes:

After completion of the course, students will be able to

1. Build an understanding of network types, architectures, topologies, networking components, communication media and techniques along with trends in networking
2. Identify data flow between two communicating hosts using various protocols at TCP/IP layers
3. Identify the role of various addresses in TCP/IP Protocol stack
4. Discover relevance of various protocols for given application

Unit I: Introduction to Computer Networks (7)

Definition, Types of Networks: Local area networks (LAN), Metropolitan area networks (MAN), Wide area networks (WAN), Wireless networks, Protocol, Design issues for the Network layers, Network Models: The OSI Reference Model, TCP/IP Model, Network Topologies. Types of Transmission Medium, Network Architectures: Client-Server, Peer To Peer, Hybrid. Network Devices: Bridge, Switch, Router, Gateway, Access Point, Modulation, Line Coding Schemes, Switching: Circuit switching, Packet switching, Multiplexing: FDM, TDM.

Unit II: Data Link Layer (8)

Introduction, functions, Design Issues, Services, Framing, Error Detection and correction, Parity Bits, Hamming Codes and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol, MAC Sub layer : Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, Introduction to Ethernet IEEE 802.3, IEEE 802.11 a/b/g/n, IEEE 802.15 Standards.

Unit III: Network Layer (10)

Functions of Network layer, Design Issues, IP Protocol: Classes of IP (Network addressing), IPv4, IPv6, Network Address Translation, Sub-netting, CIDR, Routing Algorithms: Dijkstra's, Distance vector Routing, Link State Routing, Network Layer Protocols: Address Resolution Protocol, Reverse Address Resolution Protocol, Internet Control Messaging Protocol, Routing Protocols: Routing Information Protocol, Open Shortest Path First , Border Gateway Protocol, Unicast Routing Protocols, Multicast Routing Protocol.

Unit IV: Transport Layer (7)

Transport layer design issues, Protocol Overview, Header Structure, Transmission Control Protocol (TCP) functions such as Connection Management, Error control, Flow control, Congestion control, User Datagram Protocol (UDP) overview, typical applications support, TCP Vs. UDP, TCP and UDP Socket Primitives.

Unit V: Application Layer (6)

Hyper Text Transport Protocol (HTTP): Overview, header structure, connections, request and response messages, persistence and non-persistence HTTP. Cookies, Simple Mail Transport Protocol (SMTP): Overview and Working of MIME, POP3, File Transfer Protocol (FTP): Overview and Working, identifying protocols for given application with example, Introduction to various Types of Servers, Dynamic Host Configuration Protocol (DHCP): Header, Working, Domain Name Server (DNS): Working, Proxy Server: Need and Significance, working.

Unit VI: Trends in Communication Networks (4)

Introduction to Network Security, Security mechanism, need and Services, Introduction to classical cryptography and its Type, Introduction to software defined network (SDN), Characteristics, Operations and Applications. Introduction to virtualization. (Reference from Research Papers and web)

Text Books:

1. Andrew S Tanenbaum, David J Wetherall,**'Computer Networks'**, *Pearson,* (5th Edition), (2019).
2. Forouzan B **'Data Communication and Networking'**, *Tata McGraw Hill,* (5th Edition), (2019).

Reference Books:

1. Kurose, Ross **'Computer Networking a Top Down Approach Featuring the Internet'** *Pearson,* (6th Edition), (2017).

2. Stallings W **'Data and Computer Communications'** *Prentice Hall Pvt. Ltd*. (10th Edition), (2019).
3. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff **'Unix Network Programming Volume 1'**, *Addison-Wesley Publication*, (3rd Edition), (2005).
4. Geoffrey C. Fox, Jack Dongarra, and Kai Hwang, **'Distributed and Cloud Computing'** Morgan Kaufmann, (1st Edition),(2011).
5. Stallings W, **'Cryptography and Network Security: Principles and Practice***', Pearson*, (7th Edition), (2020).

Online/Web/Other References:

1. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf
2. nptel.ac.in/courses/106/105/106105183
3. nptel.ac.in/courses/106/105/106105081
4. nptel.ac.in/courses/106/106/106106091
5. nptel.ac.in/courses/106/105/106105031

Suggestive List of Tutorials

1. Study of various networks components, devices, and cabling
2. Problem solving on Line Coding Scheme
3. Problem solving based on error control using parity code, hamming and CRC
4. Scenario based problem solving on flow control stop and wait, go back N and Selective repeat
5. Problem solving based on IP Header
6. Problem solving based on subnetting / supernetting
7. Problems based on routing algorithm.
8. Demonstration of Routing protocols on simulator
9. Problems based on TCP and UDP header
10. Case study of college network
11. Designing network for given specification
12. Research paper reading based on SDN, Network Security, Virtualization, and Satellite Network

20CE 502 Design and Analysis of Algorithms

Teaching Scheme:	**Examination Scheme**
Lecture: 3 Hrs/Week	In Semester: 50 Marks
	End Semester: 50 Marks
	Credits: 3

Prerequisite:

1. Data Structures and Algorithms (20CE302)
2. Discrete Mathematics (20CE303)

Course Objectives:

To facilitate the learners :-

1. Understand and apply methods of analysis of algorithms.
2. Learn and apply strategies for designing the algorithms.
3. Learn and apply the concept of computational complexity classes for the given problem.
4. Get acquainted with the concept of evolutionary algorithms design.

Course Outcomes:

By taking this course, the learner will be able to :–

1. Apply the knowledge of analyzing the algorithm.
2. Evaluate algorithm design techniques for solution of a problem.
3. Perceive the given problem solution from computational complexity classes point of view.
4. Build knowledge to understand the design requirements of evolutionary algorithms.

UNIT I: Introduction [7]

Basic steps to solve the problems, Performance analysis of recursive and non-recursive algorithms, Recurrences: substitution method, recursion-tree method, master method.

UNIT II: Computational Complexity Classes [7]

Basic Concepts of complexity classes, Non deterministic algorithms, The classes P and NP, NP Complete and NP Hard.

Decision problems: Clique Decision problem, Node cover Decision problem, Directed Hamiltonian Cycle Problem, Satisfiability problem, Travelling salesman problem, NP Hard problems

UNIT III: Divide and Conquer and Greedy Strategy [7]

Divide and Conquer: General Strategy, Control Abstraction, min/max problem, Binary Search, Quick Sort, Randomized quick sort and Merge Sort.

Greedy Method: General strategy, control abstraction, Knapsack problem, Job sequencing with Deadlines (Scheduling Algorithm), Minimal Spanning Tree algorithms(Graph Based Algorithm).

UNIT IV: Dynamic Programming [7]

Dynamic programming: General Strategy, Multi stage graphs, Optimal Binary Search Tree problem(OBST), 0/1 Knapsack problem, Travelling Salesperson Problem.

UNIT V: Backtracking and Branch and Bound [7]

Backtracking: General Strategy, Implicit and Explicit constraints, DFS State space tree formulation, Sum of subsets, Hamiltonian Cycle problem/Graph colouring problem, 4/8 Queens problem, Maze problem /Tower of Hanoi, 15-puzzle problem/Sudoku.

Branch and Bound: General Strategy, BFS state space tree formulation, Traveling Salesperson Problem.

UNIT VI: Introduction to Advanced Algorithms [7]

Introduction to Parallel Algorithms Matrix Multiplication/Sorting, Genetic Algorithms, Approximation Algorithms, Randomized Algorithms.

Text Books:

1. Horowitz and Sahani, "Fundamentals of Computer Algorithms", 2nd edition. Galgotia publication,, 2008, ISBN: 978 81 7371 6126
2. Gilles Brassard and Paul Bartley, "Fundamental of Algorithm.", PHI, 2010, ISBN-9788120311312 New Delhi
3. Thomas H Cormen and Charles E.L Leiserson, "Introduction to Algorithm", 3rd edition, 2009,PHI

Reference Books:

1. Fayez Gebali, “Algorithms and Parallel Computing”, Willy, 2015, ISBN 9788126553891
2. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, 2014, Pearson Education

3. A. V. Aho and J.D. Ullman, "Design and Analysis of Algorithms", Pearson Education, 2006, ISBN: 978 81 317 0205 5
4. Parag Himanshu Dave, Himanshu Bhalchandra Dave, " Design And Analysis of Algorithms", PEARSON Education, ISBN 81-7758-595-9

MOOC Courses:

1. https://nptel.ac.in/courses/106/101/106101060/ 12 weeks course offered by IIT B.
2. https://www.mooc-list.com/course/algorithms-design-and-analysis-part-1-coursera
3. https://www.cse.iitb.ac.in/~akshayss/courses/cs310-2019/index

20CE 503 Software Design and Architecture

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/Week	In Semester: 50 marks
Tutorial: 1 Hr/Week	End Semester: 50 marks
	Credits: 4

Prerequisite: Data structures (20CE 302)

Course Objectives:

To facilitate the learner to -

1. Develop familiarity with the basic concepts of software architecture and quality attributes of a system.
2. Model the software requirements of a system using Unified Modeling Language (UML) to understand the architectural, structural and behavioral aspects of the system.
3. Understand and apply various design patterns in creating an object oriented design.
4. Get exposure to the various software testing techniques and methods.

Course Outcomes:

By taking this course, the learner will be able to -

1. Analyze the concepts of software architecture and quality attributes to realize the solution of a system.
2. Build structural and behavioral models using Unified Modeling Language (UML).
3. Apply various design patterns to understand reusability in object oriented design.
4. Apply various software testing techniques at unit level, suitable to different problem areas.

Unit 1: Introduction to Software Architecture (06)

Software Development Life Cycle (SDLC), SDLC Models, Software Requirements Specification (SRS), What is Software Architecture, Why Software Architecture is important, Architectural Styles.

Unit 2: Design Using Unified Modeling Language (UML) (08)

Importance of modeling, Introduction to UML: Object-oriented modeling language, Use case Diagrams, Activity Diagrams, Class Diagrams, Sequence Diagrams.

Unit 3: Quality Attributes (08)

Understanding Quality Attributes, Quality Attribute Scenarios and Tactics - Performance, Security, Usability.

Unit 4: Creational and Structural Design Patterns (07)

What is Design Pattern, Classification of Design Patterns, Elements of Design Pattern, Creational Design Patterns - Singleton, Factory Method, Structural Design Patterns - Proxy, Adapter.

Unit 5: Behavioral Design Patterns **(06)**
Observer, Iterator, Model View Controller (MVC), Mediator.

Unit 6: Software Testing **(07)**
Introduction, Verification and Validation, V-Model, White Box testing - Structural Testing – Unit / Code functional testing, Code coverage testing, Code complexity testing, Black Box testing - Equivalence Class Partitioning, Boundary Value Analysis, Use case based testing.

Text books:

1. Len Bass, Paul Clements, Rick Kazman, **'Software Architecture in Practice'**, *Pearson Education,* (3rd Edition)(2013).
2. Grady Booch, James Rumbaugh, Ivar Jacobson, **'The Unified Modeling Language User Guide'**, *Pearson Education,* (2nd edition)(2008).
3. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, '**Design Patterns- Elements of Reusable Object-Oriented Software',** *Pearson Education,* (2002).
4. Srinivasan Desikan, Gopalaswamy Ramesh, '**Software Testing Principles and Practices'**, *Pearson Education*, ISBN 81-7758-121-X (2013).

Reference books:

1. Len Bass, Paul Clements, Rick Kazman, **'Software Architecture in Practice'**, *Pearson Education,* (2nd Edition) (2006).
2. Mary Shaw and David Garlan, **'Software Architecture – Perspectives on an Emerging Discipline'**, *Prentice Hall of India,* (1996).
3. Richard N. Taylor, Nenad M. and Eric M. Dashofy, **'Software Architecture: Foundations, Theory and Practice'**, *Wiley,* (2006).
4. Jim Arlow and Ila Neustadt, **'UML 2 and the Unified Process –Practical Object-Oriented Analysis and Design',** *Pearson Education*, (2nd edition) (2006).
5. Iien Burnstein, '**Practical Software Testing',** *Springer (India) private limited,* (2005).

Tutorials - Preamble:

The scope of tutorials for "Software Design and Architecture" includes exercises based on requirements capturing, analysis, design and testing of sample applications. During tutorials, problem solving and system design skills of students are challenged and improved. For a chosen hypothetical system, students are expected to identify its scope, prepare SRS document, build analysis/design level UML models and identify the test cases. The students are also expected to analyze the quality attributes requirements for the chosen system and elaborate the same using quality attributes scenarios. The following is a sample list of tutorials, covering the various concepts in the course. The objective of tutorials is to provide an opportunity for students to explore as per their interests. Consequently, these tutorial statements will be further detailed during conduction, according to the scenarios under consideration.

Example List of Tutorials:

1. Study architectural styles and submit a report on these styles.
2. A case study of any website or any other large system and its architecture for quality attributes requirements such as Performance, Security, Usability and Availability.
3. Design a Software Requirement Specification (SRS) document for a given system.
4. Draw Use case diagrams for capturing and representing requirements of a given system.
5. Draw Activity diagrams to display the business flows for a given system.
6. Draw Class diagrams to identify and describe key concepts like classes, relationships and other classifiers like interfaces.
7. Draw Sequence diagrams to show message exchanges in a given system.
8. Identify suitable design patterns for a given application.
9. Apply various Black Box testing methods for unit testing of a sample application.
10. Apply various White Box testing methods for unit testing of a sample application.

20CE 501L Computer Networks Laboratory

Teaching Scheme	**Examination Scheme**
Practical: 2 Hours / Week	In Semester: 25 Marks Practical : 25 Marks Credits: 1

Prerequisite:

Course Objectives:

To facilitate the learner to

1. Learn computer network topologies and types of network
2. Use modern tools for network traffic analysis and various networking configurations.
3. To learn network programming.
4. To develop an understanding of various protocols, modern technologies and applications

Course Outcomes:

After completion of the course, students will be able to

1. Demonstrate error control, flow control techniques and analyze them
2. Configure switches and routers.
3. Demonstrate LAN and WAN protocol behaviour using Modern Tools
4. Develop Client-Server architectures and prototypes

Preamble:

A large part of the lab would be for understanding the concepts of Computer Networking. Assignment statements are in brief and should be implemented in JAVA/Python programming language along with packet tracer and wire shark tool. Motivation here is that students should be able to code the basic algorithm and also should be able to make use of built-in functions available in different libraries of Java/Python. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, logic of solution. Students will be encouraged to solve open problems in different domains. Faculty will appropriately adopt assignments on similar lines as the examples shown here. Group A assignments consisting of simulation of static and dynamic routing using cisco packet tracer tool. Group B assignment comprises socket programming to demonstrate process to process communication. Group C assignment consists of case study implementation.

Suggestive List of Assignments

Group A : (Mandatory)

1 Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.

2 Design an IP scheme for a WAN network (minimum 3 networks) using Cisco Packet Tracer tool (Static Routing).

3 Setup a WAN which contains wired as well as wireless LAN by using a packet tracer tool. Demonstrate transfer of a packet from LAN 1 (wired LAN) to LAN2 (Wireless LAN).

4 Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.

5 Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.

Group B: (Any Two)

1 Use packet Tracer tool for configuration of 3 router network using one of the following protocol RIP/OSPF

2 Write a program using TCP socket for wired network for following a. Say Hello to Each other b. File transfer c. Calculator

3 Write a program for DNS lookup. Given an IP address as input, it should return URL and vice versa.

4 Write a program to demonstrate Sub-netting and find subnet masks.

5 Configuring Ftp server for file upload /download using Cisco Packet Tracer.

Group C

1 Create a network for N (e.g. More than 10) users all user should get the concurrent internet connectivity.

2 Create and configure Virtual Machine

Virtual Laboratory: http://vlabs.iitb.ac.in/vlab

20CE 504L Programming Skills Development II Laboratory

Teaching Scheme
Practical: 4 Hours / Week

Examination Scheme
In Semester: 25 Marks
Oral : 25 Marks
Credits: 2

Prerequisite:

1. Fundamentals of Programming Languages – II (20ES05)
2. Data Structures and Algorithms-II (20CE305)
3. Programming Skills Development-I Laboratory (20CE 306)

Course Objectives:
To facilitate the learner to

1. Explore the usage of mobile development tools.
2. Learn the process of development of mobile application
3. Create data-driven mobile applications
4. Create mini project using all the concepts

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

1. Select suitable configuration parameters, components, API's, libraries for mobile application development.
2. Design an android application using widgets, layouts, event handlers, intents etc.
3. Develop mobile application using advance features of android like database, multimedia, canvas, graphics etc.
4. Create a small mobile application.

Preamble:

The lab would be for understanding the syntax and semantics of Android programming and implementation of some real-world simple applications. Assignment statements are in brief and should be implemented using android studio.

Motivation here is that students should be able learn the App development and also should be able to analyze problems and select suitable built-in tools/API.

Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, constructs and design of simple applications. Students will be encouraged to solve open ended problems in different domains. Faculty will appropriately adopt assignments on similar lines as the examples shown here.

Group A assignments are for learning the basics of android programming

Group B assignments are for learning advanced features like fragments, custom views, animations, Sensor capabilities to the App. .Group C assignment is open ended application development

Suggestive List of Assignments

Group A : (Mandatory)

1. Download, install and configure android development tools, plugins and SDK / Studio
2. Develop a simple Applications like
   a. Calculator
   b. Unit Converter (Scale,Temperature)
   c. BMI using UI Widgets – button, textview, editview etc.
3. Develop an application that uses Spinner component
4. Develop an application that uses intent, event listener and different Layout Managers
5. Develop an application that draws basic graphical primitives on the screen
6. Develop an application that make use of Alert Dialog

Group B: (Any Three)

1. Develop a mobile application that makes use of database for student/employment/ Library
2. Develop a mobile application that makes use of Uri,XMlPullparser like RSS feed
3. Develop a mobile application that implements Multi threading
4. Develop a mobile application that creates an alert upon receiving a message.
5. Develop a mobile application that uses time picker,alarm,adapter like creating alarm clock
6. Develop a mobile application for multimedia Application
7. Develop a mobile application for image transformations like Translation,Scaling and rotation
8. Develop an application that reads/writes data to-from the SD card.
9. Develop an application that uses contact class and methods like send email/SMS
10. Develop a sensor based application using Motion sensors,Position sensors, Environmental sensor

Group C

1. Mini Project

20PECE 501A Digital Image Processing

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hours / Week	In Semester: 50 Marks
	End Semester: 50 Marks
	Credits: 3

Course Objectives:

To facilitate the learner to

1. Understand basic concepts of digital image processing.
2. Learn and apply image enhancement and Image Segmentation techniques.
3. Understand object Recognition, Image Restoration and reconstructions.
4. Learn and apply image compression techniques and Understand image processing applications.

Course Outcomes:

After completion of the course, students will be able to

1. Apply basic steps of digital image processing on given images
2. Select the image enhancement techniques.
3. Make use of Image Restoration, reconstructions techniques.
4. Identify the image compression techniques.
5. Choose Image Segmentation techniques for given images.

(07)

Unit I: Introduction to Image Processing

Introduction to digital image processing: Origin, usage and application of image processing, Fundamental steps and component of image processing system,representation of digital images. Basic relationships between pixels, introduction to Human Visual System,Image sensing and acquisition,Basic concepts in sampling and quantization,Basic operations:Convolution,Arithmetic and Logical Operations.

Unit II: Image Enhancement Techniques (08)

Basic image preprocessing (contrast enhancement, simple noise reduction) some basic gray level transformations,Histogram Processing-Histogram Equalization,Histogram stretching,Spatial filtering- Smoothing and Sharpening Spatial filters. Frequency Domain: Introduction to Fourier Transform– frequency domain filters.

Unit III: Image Compression (07)

Introduction to Image Compression and its need, Coding Redundancy,Classification of Compression Techniques (Lossy and Losless - JPEG,RLE, Huffman, Shannon fano,Arithmetic coding), Scalar & Vector Quantization.

Unit IV: Image Restoration & Reconstruction (06)

Model of Image degradation, Noise Models,Classification of image restoration techniques,Order Statistics fileters-Mean,Median,Min,Max,Alpha trimmed mean filter,Geometric and harmonic mean filter,Inverse filtering, Wiener filtering,Blind-deconvolution techniques.

Unit V: Image Segmentation, Analysis and Object Recognition (08)

Point detection, Lines detection,Edge detection,Classification of image segmentation techniques: Edge-based Segmentation, Region based techniques. Binarization: Global Thresholding, Adaptive thresholding. Types of Edge detector: derivative filters-Prewitt, Sobel, Canny.

Introduction to Object Recognition, Object Representation(Signatures, Boundary Skeleton), Simple Boundary Descriptors, Regional descriptors- Topological feature(Texture).

Morphological Operations:Basics of Set Theory; Dilation and Erosion - Dilation,Erosion; Structuring Element; Opening and Closing;

Unit VI: Advances in Image processing Applications (06)

Medical Image Processing, Remote Sensing, Synthetic-aperture radar (SAR) Image Processing,Image registration,Biometric Authentication Methods like Face detection, Iris Recognition.

Text Books:

1. R.C. Gonzalez, R.R. Woods, '**Digital Image Processing**', ISBN 978-81-317-2695-2,*Person (*Third Edition) ,(2011)
2. Sridhar S. '**Digital Image Processing',** *Oxford University Press, (Second Edition),(2016)*
3. S.Jayaraman, S. Esakkirajan,T. Veerakumar , **'Digital Image processing',**ISBN 978-0-07-014479-8, *Mcgraw Hills Publication (Tenth reprint),(2013)*

Reference Books:

1. Sonka, Hlavac, Boyle, '**Digital Image Processing and Computer Vision',** ISBN 978-81-315-0555-7,*Cenage Learning* (Sixth Indian Reprint) ,(2011)
2. B. Chanda, D.Datta Mujumdar '**Digital Image Processing And Analysis',** *PHI*, ISBN 978- 81-203- 4325-2, (Second Edition),(2013)
3. Anil Jain, '**Fundamentals of Digital Image Processing**', *PHI*, ISBN-81-203-0929-4 (Indian Reprint) ,(1995)
4. Basudeb Bhatta ' **Remote Sensing and GIS'** *Oxford University Press*, ISBN 978-0-19-807239-3 (Second Edition)(2014)

Online/Web/Other References:

1. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6504845
2. https://searchsecurity.techtarget.com/definition/biometric-authentication

20PECE 501B Java Full Stack Technologies

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/week	In Semester: 50 marks
	End Semester: 50 marks
	Credits: 3

Prerequisites: Data Structures (20CE 302)

Course Objectives:
To facilitate the learner to -

1. Get exposure to full stack development in Java technologies.
2. Develop familiarity with the client side Java technologies.
3. Gain comprehensive knowledge about Java server side technologies for enterprise application development in practice.
4. Get familiar with the web services based approach for real-life application development.
5. Get acquainted with the database development technologies in Java.

Course Outcomes:
By taking this course, the learner will be able to -

1. Choose suitable client side Java technologies.
2. Analyze Java server side technologies for enterprise application development.
3. Analyze the characteristics of web services paradigm.
4. Analyze the role of Java database development technologies to realize their suitability for application development.

Unit 1: Client Side Web Technologies (07)

n-tier architecture, HTTP request - response, Web browser, HTML, CSS, XML, JSON, JavaScript (JS), Document Object Model (DOM), Introduction to jQuery, Asynchronous JavaScript And XML (AJAX).

Unit 2: Server Side Java Web Technologies (07)

Introduction to server side technology, Common Gate Interface (CGI), Java Servlets, Java Server Pages (JSP), Session tracking, JSP tags, Java Beans, MVC architecture.

Unit 3: ReactJS (06)

Overview of ReactJS: Introduction, Features, Advantages, Comparison with AngularJS, Introduction to Nodejs; ReactJS concepts like components, virtual DOM, JSX and APIs.

Unit 4: Java 2 Enterprise Edition (J2EE) Technologies (08)

Introduction to J2EE technologies, Enterprise Java Beans (EJB), Java Messaging Service (JMS), Remote Method Invocation (RMI).

Unit 5: Java Web Services (07)

Web Services: Overview; Service Oriented Architecture (SOA), Java Web services based on SOAP and REST, Java Web services API for SOAP and REST based web services:
JAX-WS, JAX-RS.

Unit 6: Java Database Programming and Hibernate (07)

Java Database Connectivity (JDBC), Java Transaction API (JTA), Java Persistence API (JPA), Hibernate: Overview, architecture, Object Relational (OR) Mapping.

Text books:

1. Kogent Learning Solutions Inc., **'Web Technologies: HTML, JS, PHP, Java, JSP, ASP.NET, XML, AJAX, Black Book'**, *DreamTech Press*, ISBN: 978-81-7722-997-4, (2015).
2. Kogent Learning Solutions Inc., **'Java Sever Programming Java EE6 Black Book'**, *DreamTech Press*, ISBN: 978-81-7722-936-3, (2013).
3. Stoyan Stefanov, **'React - Up & Running: Building Web Applications',** *O'Reilly,* ISBN: 9781491931820, (2016).
4. William Crawford, Jim Farley, **'Java Enterprise in a Nutshell'**, *O'Reilly*, ISBN-13: 978-0596101428, 3rd Edition, (2005).

References books:

1. Mark Tielens Thomas, **'React in Action'**, *Manning Publications*, ISBN: 978-1617293856, (2018).
2. Kevin Mukhar, Chris Zelenak, James L. Weaver and Jim Crume, **'Beginning Java EE5: From Novice to Professional'**, *Apress*, ISBN-13: 978-8181284020, (2006).
3. Kirupa Chinnathambi, **'Learning React: A Hands-on Guide to Building Web Applications Using React and Redux',** *Addison Wesley,* (2016).
4. Jim Keogh, **'The Complete Reference J2EE'**, *McGraw Hill Education*, ISBN: 978-0-07-052912-0, (2012).

Web References:

1. https://learn.jquery.com
2. https://docs.oracle.com/javaee/7/tutorial/
3. https://reactjs.org

20PECE 501C Statistics for Computer Science

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hours / Week	In Semester: 50 Marks
	End Semester: 50 Marks
	Credits: 3

Course Objectives:

To facilitate the learner to

1. Utilize fundamentals of statistics and descriptive statistics concepts.
2. Analyse data using correlation, regression and multivariate analysis.
3. Apply statistical inference techniques for dealing with uncertainty in decision making.
4. Apply analysis of variance technique to check how different the samples are from each other.
5. Apply statistics concepts in different applications.

Course Outcomes:

After completion of the course, students will be able to

1. Apply the methods of descriptive statistics on different types of data.
2. Experiment with statistical analysis and multivariate analysis using correlation and regression.
3. Make use of sample inferential statistics to draw inference.
4. Perform analysis of variance for groups of data.
5. Apply the statistics concepts in applications such as manufacturing, economics, business analysis and forecasting.

Unit I: Basic statistics (10)

Definition, collection and type of data, processing of data, classification, tabulation and graphical representation of data, limitation of statistics.

Types of averages: arithmetic mean, median, mode, geometric mean, harmonic mean, relationship among averages, variation, merits and limitations of variation, standard deviation.

Unit II: Correlation and Regression (7)

Introduction, types of correlation, methods of studying correlation: scatter diagram, graphic method, Karl Pearson's coefficient of correlation, Rank correlation coefficient

Regression analysis: Introduction, uses of regression analysis, difference between correlation and regression analysis. Regression lines, regression equations, regression coefficient and its properties.

Unit III: Multivariate Analysis (4)

Partial regression, partial correlation, multiple correlation, multivariate regression, principal component analysis (PCA).

Unit IV: Statistical Inference -Test of Hypothesis (7)

Introduction, procedure of testing hypothesis, types of hypothesis, two types of error in testing of hypothesis, two-tailed and one-tailed test t-test, chi-square test, F-test, degrees of freedom, relation between t-test, chi-square and F-test.

Unit V: Analysis of Variance (6)

Introduction, assumptions and techniques of analysis of variance, One-Factor analysis of variance, two factor analysis of variance: Parameter estimation and testing hypotheses.

Unit VI: Applications of Statistics (8)

Introduction to statistical quality control, acceptance sampling, Introduction to business forecasting, Introduction to index numbers for economic and business analysis.

Text Books:

1. "Statistical Methods", S.P. Gupta, 41st Edition, 2011, ISBN :978-81-8054-862-8, Sultan Chand and Sons publication.
2. "Basic statistics", B.L. Agarwal, 9th Edition, 2011, ISBN:978-81-224-2472-0, New Age publication.
3. "Statistics in Nutshell", Sarah Boslaugh and Paul Andrew Watters, 2008, ISBN: 978-81-8404-568-0, SPD O'Reilly publication.

Reference Books:

1. "Statistical Data analytic" by Piegorsch W.W., 2017, ISBN:978-81-324-2472-4, Wiley publication
2. "Introductory statistics", Sheldon M. Ross, 2nd Edition, 2006, ISBN: 81312-00485, Elsevier publication.
3. "Applied multivariate statistical analysis", Richard A. Johnson, Dean W. Wichern, 6th edition, 2012, ISBN-978-81-203-4587-4, PHI Learning

Online/Web/Other References:

1. Nptel, coursera courses on probability and statistics

20PECE 501D Linux Internals

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hours / Week	In Semester: 50 Marks
	End Semester: 50 Marks
	Credits: 3

Course Objectives:

To facilitate the learner to

1. Understand basic concepts of UNIX Operating System and booting process.
2. Understand Linux Process and threads.
3. Understand memory management in Linux.
4. Learn basics of Inter process communication with respect to Linux.
5. Understand use of AWK scripting and MAKE tool in Linux.
6. Learn advance concepts in Linux operating system.

Course Outcomes:

After completion of the course, students will be able to

1. Apply the basic knowledge about UNIX operating system.
2. Build basics concepts in Linux process and thread management.
3. Make use of Linux Memory management concepts.
4. Utilize the concepts of inter-process communication.
5. Choose AWK scripting and MAKE tool for Linux programming.
6. Explore the advancements in Linux Operating Systems.

Unit I: Fundamental of UNIX operating system (07)

Introduction, UNIX Operating Systems structure, Kernel architecture, types of kernel, Operating system: Booting process, Grub I, Grub II, Representation of files, Systems Call File system, Free Space Management, Disk management, Concept of Buffer management in UNIX /Linux.

Unit II: Process and threads in Linux (08)

Process states and transitions, layout of system memory, Context of a process, saving the context of a process, Process creation, Signals, Process termination, Concept of threads, Linux processes and thread management, Introduction to threads , Thread libraries, Thread issues, Multithreading models, Process management and Linux scheduler.

Unit III: Memory management and virtual memory in Linux (07)

Swapping, Demand Paging, A hybrid system with swapping and demand paging, memory management requirements, Memory partitioning, Security Issues in memory management, Linux memory management.

Unit IV: Inter-process Communication in Linux (07)

Process tracing, System V IPC, Network communication, sockets, Multiprocessor systems: problem with multiprocessor systems, solution with master slave processes, Linux Inter process communication: User level IPC mechanism, Kernel synchronization, socket programming.

Unit V: Tools and Technologies (07)

Search and Sort tools: grep, egrep, fgrep, MAKE tool: When to use MAKE, Macros, abstractions and shortcuts, make, nmake, cmake. AWK tool: AWK syntax, AWK grammar, AWK scripting, Linux Utilities, Version Control Systems for Linux.

Unit VI: Variants in Linux (06)

Hand-held systems: requirements, Linux as hand-held operating system, Linux for distributed systems, technology overview, Case-study of Google Android, Linux in Supercomputing, Linux and Cloud, Linux in Business, Linux Container and Virtual Machines.

Text Books:

1. Maurice J. Bach, "The Design of the Unix Operating System", Third Edition, 2013, Pearson, ISBN 978-81-203-0516-8.
2. Pramod Chandra P. Bhatt, "An introduction to Operating Systems: Concepts and Practice (GNU/Linux)", PHI, (Fourth edition), (2014), ISBN-978-81-203-4836-3.
3. Evi Nemeth, Garth Snyder, Tren Hein, Ben Whaley, "Unix and Linux System Administration Handbook", Pearson, (Fourth Edition), ISBN: 978-81-317-6177-(2014).

Reference Books:

1. William Stallings, "Operating System-Internals and Design Principles", Prentice Hall India, ISBN-81-297-0 1 094-3.
2. David Rusling, "The Linux Kernel", Addison Wesley, (Second edition), ISBN 978-0201770605.
3. Sumitabha Das, "UNIX Concepts and Applications", ISBN 0-07-053475-6.

20PECE 501LA Digital Image Processing Laboratory

Teaching Scheme
Practical: 2 Hours / Week

Examination Scheme
In Semester: 25 Marks
Oral : 25 Marks
Credits: 1

Prerequisite:

Course Objectives:

To facilitate the learner to

1. Learn Basics Image Processing operations like image Read, Write, Add, subtract
2. Understand and apply algorithms used for image enhancement, edge detection
3. Able to develop an Image Processing application using various techniques
4. Learn and use different Image Processing Tools

Course Outcomes:

After completion of the course, students will be able to

1. Apply basic operations on given image
2. Apply effectively algorithms for image enhancement, edge detection
3. Develop small image processing application using various techniques
4. Make use of Image Processing Tool

Preamble:

The lab would be for understanding the basic concepts of image processing and implementation of some real-world simple applications. Assignment statements are in brief and should be implemented Using Opencv in Python/JAVA programming language. Motivation here is that students should be able to code the basic algorithm and also should be able to make use of built-in functions available in different libraries of Opencv and Python/Java. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, logic of solution and simple application. Students will be encouraged to solve open problems in different domains. Faculty will appropriately adopt assignments on similar lines as the examples shown here.

Group A assignments are on capturing images and performing different manipulation on image data such as arithmetic and logical operations, to improve the quality of image using different enhancement techniques. Group B assignments are on filtering techniques and Group C assignment is open ended application development

Suggestive List of Assignments

Group A : (Mandatory)

1. Study different file formats and Write a program to create a simple image file in .tiff format, and display it .
2. Write a program to perform Arithmetic operations/Logical operations
3. Write a program to perform Intensity Transformation technique on given image
4. Write a program for image enhancement techniques using Histogram

Group B: (Any Three)

1. Write a program using derivative filtering technique for Edge detection
2. Write a program to illustrate Morphological transformation
3. Write a program to illustrate Image Restoration techniques
4. Write a program using Edge detectors for Edge detection
5. Write a program for Non-Linear filtering using convolutional masks- effects of a median filter on an image corrupted with impulsive noise.

Group C

1. Implement a small Image processing application using MATLAB/ OpenCV

20PECE 501LB Java Full Stack Technologies Laboratory

Teaching Scheme	**Examination Scheme**
Practical: 2 Hrs/week	In Semester: 25 Marks
	Oral : 25 Marks
	Credits: 1

Course Objectives:
To facilitate the learners to -

1. Understand the Installation and Configuration setting related aspects of web server, integrated development environments and various frameworks, in the development of web applications.
2. Understand the role of various technologies used for real-life application development.
3. Get exposure to full stack development in Java which includes client side and server side technologies, web services and database development technologies.
4. Gain practical knowledge about the various client side and Java server side technologies for application development in practice.

Course Outcomes:
By taking this course, the learner will be able to -

1. Make use of suitable client side Java technologies.
2. Experiment with various Java server side technologies like Java Servlets, Java Server Pages, Web services and JPA for web application development.
3. Make use of Java Sockets library and Java RMI framework for the development of sample client-server applications.
4. Build a sample web application using suitable technologies at various tiers.

Preamble:

Development of web applications need technologies at various levels, which play different roles in the overall web architecture. The intent of Java Full Stack Technologies Laboratory is to enable the understanding of the role of various technologies in full stack development and implementation of some real world application scenarios using these technologies. Assignment statements are in brief and should be implemented with Java web technologies. Motivation here is that students should be able to develop the user interface, business logic and the database programming parts of a typical web application, using the APIs/libraries provided by various client side and server side Java technologies. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, logic of solution and simple application. Students will be encouraged to build solutions for real world business scenarios in different domains, to fulfil the end-user requirements. Faculty will appropriately adopt assignments on similar lines as the examples shown here. Group A assignments are on applying various client side technologies and basic server side technologies. Group B assignments are on exploring the use of technologies like EJB, Web services and Hibernate. Group C assignment is on the development of sample web application.

Suggestive List of Assignments:

Group A: (Mandatory)

1. Develop dynamic and interactive web client using HTML, CSS and JavaScript technologies. Make use of these technologies to develop suitable web forms, layout and to perform validation of form data, for this web client.

   Sample application scenario:
   Consider that a student needs to register for an online course portal. For this scenario, develop an HTML form for "Course Registration", make use of CSS for layout design of this form and perform validation on various fields of this form using JavaScript.

2. Develop dynamic and interactive web client using XML and AJAX technologies, to enable rich user experience.

   Sample application scenario for AJAX:
   Consider a web form for an administrator of an "Online Shopping Application". An administrator can select the name of a customer from the drop down list box on the web form. Then on the same page, the details of the customer such as shipping address should get displayed.

3. Develop dynamic and interactive web client using jQuery as a client side JavaScript library. For this web client, implement event handling and animation effects using jQuery.
4. Develop dynamic and interactive web client using ReactJS as a client side library. Make use of various features of ReactJS such as components, APIs etc.
5. Implement a simple client-server application like echo server or chat server using Java Sockets. Make use of multithreading in Java for handling the requests from multiple clients.
6. Implement an application using Java RMI to understand distributed application environment. The remote object accesses database using JDBC.
7. Implement a sample web application scenario using Java Servlets, Java Server Pages and Java Beans as the server side dynamic content generation technologies. Make use of MVC architecture for this implementation and also show the appropriate usage of the various capabilities of these technologies such as session tracking, tag library, implicit objects, directives etc.

   Sample application scenario:
   Consider a simple web form where you give Student Roll number and get back Student Profile details from the database. Make use of MVC architecture, based on Java Servlet, JSP and Java Bean to implement this web based scenario.

Group B: (Any One)

1. Implement a sample EJB based scenario for any application like online movie ticket booking, online college admission portal, online railway reservation etc. Make use of various types of beans such as session beans and entity beans, for the implementation of business methods and persistence of data.
2. For a sample application scenario, implement and consume the suitable web services using SOAP or REST protocol.
3. Make use of JPA with Hibernate framework for performing the create, retrieve, update and delete (CRUD) operations on the backend database.

   Sample application scenario:
   Consider "Course Information Management" as a typical Database Application. This application may have database tables like Courses, Participants etc. Make use of JPA with Hibernate framework to access the data from the Courses table in this above application.

Group C:

1. Design and develop a typical web application like online cab booking, online food ordering application, online tours and travel portal etc. For the development of this application, choose the appropriate technologies for the client side aspects, server side business logic and database development.

20PECE 501LC Statistics for Computer Science Laboratory

Teaching Scheme	**Examination Scheme**
Practical : 2 Hours / Week	In Semester: 25 Marks
	Oral: 25 Marks
	Credits: 1

Prerequisite: NA

Course Objectives:

To facilitate the learner to

1. Understand and use the basic statistical tool for statistical operations and interpretation of data.
2. Use knowledge of data representation for given data points.
3. Apply correlation, regression model and ANOVA model for given data.
4. Apply hypothesis testing to draw conclusions for given data.
5. Use concepts of statistics for real life problems.

Course Outcomes:

After completion of the course, students will be able to

1. Perform basic statistical operations on given data using statistical programming and tools.
2. Apply different data representation methods for interpretation of given data.
3. Apply various models of regression, correlation and ANOVA to predict and find relation between given data.
4. Apply hypothesis testing to draw inference for given data.
5. Develop small statistical applications using different statistical techniques.

Preamble

A large part of the lab would be for understanding the basic concepts of statistics and implementation of some real-world simple applications. Assignment statements are in brief and should be implemented in R/Python programming language. Motivation here is that students should be able to code the basic statistical techniques on given data and also should be able to make use of built-in functions available in different libraries of R/Python. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, logic of solution and simple application. Students will be encouraged to solve open problems in different domains. Faculty will appropriately adopt assignments on similar lines as the examples shown here. Assignments can be done on any data dataset. Students and faculty are advised to consider different datasets.

Suggestive List of Assignments

Group A: (Mandatory)

1.Getting started with software, installation, its objects and data types
2.Graphical presentation of data in different plot forms/diagrams using software tool
3.Apply basic statistical operations, measure of location (Arithmetic mean, harmonic mean, geometric mean, median, mode)
4.Perform measure of dispersion, standard deviation, quartile deviation etc.

Group B: (Any Four)

1.Plot the diagram for the given data, develop the regression model that best describes the data, and also predict output for the given value.
2.Perform correlation analysis (positive, negative, zero) that describes the degree to which variables are linearly related to each other.
3.Perform test of hypothesis, one sample t-test, paired t-test on given data and see how to use them for statistical inference.
4.Perform test of hypothesis, chi-squared goodness of fit test, on given data and see how to use them for statistical inference.
5. Perform analysis of variance (ANOVA) on data for evaluating hypothesis.

Group C

Data analysis case study for readily available data set using the statistical techniques studied. Take a dataset freely available, apply statistical analysis technique or statistical machine learning technique on that dataset, also graphically represents the results.

20PECE 501LD Linux Internals Laboratory

Teaching Scheme	**Examination Scheme**
Practical: 2 Hours / Week	In-Semester: 25 Marks
	Oral: 25 Marks
	Credits: 1

Prerequisite:

- Operating Systems Laboratory (20CE 407)

Course Objectives:

To facilitate the learner to

1. Understand basic commands of LINUX Operating System.
2. Understand and write shell script for a given task.
3. Apply socket programming concepts.
4. Learn use of AWK programming and MAKE tool.
5. Apply Inter process communication concepts for solving a problem.

Course Outcomes:

After completion of the course, students will be able to

1. Apply fundamental concepts of LINUX Operating System.
2. Implement shell script for a given problem statement.
3. Build the concepts for socket programming.
4. Explore AWK script and MAKE tool to write basic programs.
5. Make use of Inter-process communication concepts for solving a problem.

Preamble:

Linux Internals Laboratory is designed to understand and implement the concepts of Linux operating System. Motivation here is that students should be able to code the shell script and make use of inter-process communications concepts. Assignments can be framed and expanded in such a way that it enhances the concepts and logic of the solution. Students will be encouraged to solve open problems using different tools like MAKE, AWK script. Group A assignments are based on Linux commands, shell script and inter-process communication using pipes. Group B assignments are on AWK script, MAKE tool and Sockets programming. Group C assignment is to explore Linux Environment in more advanced way.

Suggestive List of Assignments

Group A : (Mandatory)

1.Exploration of Linux commands - File handling utilities, Process utilities, Disk utilities, Networking commands, Text Processing utilities and Back-up utilities.

2. Write a program for counting characters, words or lines and searching text file contents using grep command.
3. Write a program for Inter-process communications between related processes using pipes.

4. Write a Java program to implement the following UNIX commands using following system calls:

- cat
- ls
- mv

5. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.

Group B: (Any Four)
7. Write an AWK script to count the number of lines in a file that do not contain vowels.

8. Write a program using the MAKE tool to compile all java source files that need to be re-built.
9. Write a program in Java/Python to create a RAMDRIVE and associate an acyclic directory structure to it.
10. Write a Java program to create Child process (using fork), Zombie, Orphan and displaying system information.
11. Write a AWK script to filter, split a line into fields, printf, variables and expressions, the comparison operator, number processing.
12. Write an IPC program using pipe. Process A accepts a character string and Process B inverses the string. Pipe is used to establish communication between A and B processes using Python or Java.

Group C (Any One)
1. Write a program to implement Linux installation commands/administrative commands / security commands/version controlling commands/IPC Commands/ File Management Commands.
2. Write client and server programs for interaction between server and client processes using Unix Domain sockets and Internet Domain Sockets.