R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

Semester-V(Third Year)

S.No	Course Code	Course Title	Hours Per Week		Scheme of Examination			Category
			L	P	Internal Marks	Sem End Exam Marks	Credits	
1	CO/CS/IT/CM311	Automata Theory &Formal Languages	3	0	30	70	3	PC
2	CO 312	Software Engineering	3	0	30	70	3	PC
3	CO/CS/IT 313	Design And Analysis of Algorithms	3	0	30	70	3	PC
4	CO 314-**COEL01**	**IoT with Cloud Computing** (Professional Elective–I)	3	0	30	70	3	PE
5	CO 315-**JOEL24**	**Artificial Intelligence** (Job-Oriented Elective–I)	3	0	30	70	3	OE
6	CO351	Software Computing Lab	0	3	30	70	1.5	PC
7	CO352	**IoT with Cloud Computing Lab** (Professional Elective Lab)	0	3	30	70	1.5	PE
8	CO/CS/IT/CM/CB 353	Summer Internship	-	-	100	-	1.5	PR
9	COSL3	**Soft Skills** (Skill Oriented Course–III)	1	2	100	-	2	SC
		TOTAL	**16**	**8**	**410**	**490**	**21.5**	

Category	CREDITS
Professional Core Courses	12
Professional Elective Course	3
Open Elective Course/Job Oriented Elective	3
Soft Skills-Skill Oriented Course*	2
Summer Internship	1.5
TOTALCREDITS	**21.5**

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

CO/CS/IT/CM 311 **AUTOMATA THEORY & FORMAL LANGUAGES** **L P C**
3 0 3

Course Objectives:

The main objectives of this course are to:

1. Introduce the types of Finite Automata and properties of Regular Expressions.
2. Explain Context-Free Grammars and Push Down Automata
3. Introduce the Turing Machine and explain undecidability concept.

Course Outcomes:

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

1. Explain the fundamental concepts of Automata and Formal languages. L2
2. Apply the knowledge of Automata Theory, Formal languages, Grammars Regular Expressions for solving various problems. L3
3. Design PDAs for various languages. L4
4. Design Turing machines to solve problems.

Course Content:

UNIT I **CO-1** **12 Periods**

Automata: Introduction to Automata, The central concepts of automata theory - Alphabets, Strings, Languages.

Finite Automata: An Informal picture of finite automata, Deterministic finite automata (DFA) - Definition of DFA, DFA processing strings, Notations for DFA, Extended transition function, the language of DFA, Non-deterministic NFA Finite.

Automata with $\in$ transitions: Use of $\in$ - transition, notation for an $\in$ - NFA, Epsilon closures, extended transitions and languages, Applications.

UNIT II **CO2** **12 Periods**

Regular Expressions and Languages: Regular expressions, finite automata and regular expressions, Algebraic laws of regular expressions.

Properties of Regular Languages: Proving languages are not regular – Pumping lemma for regular languages, Applications of the pumping lemma, Closure Properties of Regular Languages, Equivalence and minimization of automata – Minimization of DFA

UNIT III **CO 3** **12 Periods**

(Construction based treatment & proofs are excluded)

Context Free Grammars: Context Free Grammars, Parse Trees, Constructing parse trees, derivations and parse trees, ambiguous grammars.

Pushdown Automata: Definition of the Pushdown automata, the languages of P DA , Equivalences of PDA's and CFG's.

Context free languages: Normal form's for context- Free grammars, the pumping lemma for context free languages.

UNIT IV **CO 4** **12 Periods**

Properties of Context free languages: closure properties for context free languages, Decision properties for CFL's.

Introduction to Turing Machines: The Turing Machine, programming techniques for Turing machines.

Undecidability: A language that is not recursively enumerable, an undecidable problemthat is RE, Undecidability problems about TM, Post's Correspondence problem.

Learning Resources:

Text Book:

1. John. E.Hopcroft, R.Motwani, & Jeffery.D Ullman, "Introduction to Automata Theory, Languages and Computations", Second Edition, Pearson Education, 2003 .

Reference Books:

1. Daniel I.A.Cohen, 'Computer Theory',
2. KLP Mishra & N.Chandrasekharan, 'Theory of Computation', PHI.
3. Micheal Sipser, "Introduction of the Theory and Computation", Thomson Brokecole, 1997.
4. R.K.Ragade, "Automata and Theoretical Computer Science", First Edition, Pearson Education, 2004

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

CO 312 **Software Engineering** **L P C**
3 0 3

Course Objectives:

At the end of the course, the student will understand and

1. Acquire knowledge on the principles and process models for software development.
2. Explain the specific requirements for a given software project
3. Acquire knowledge on design concepts and user interface principles for Software development
4. Examine various testing techniques and metrics applicable to a Software project
5.

Course Outcomes:

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

CO1. Describe the software engineering process model required to create a software system.
CO2. Discuss the software requirements and analyze a model for a software project.
CO3. Design and specify software components for real-world problems.
CO4. Evaluate various software testing techniques and metrics.

Course Content:

UNIT I **CO-1** **14 periods**

Software and Software engineering: The Nature of Software, Defining Software, Software Application Domains, Legacy Software, The software Process.

The Software Process: Process Models: A Generic Process Model, defining a Framework Activity, identifying a task set, Process Assessment and Improvement, Prescriptive Process Models: The waterfall model, Prototyping Process model, Evolutionary process model, The Unified Process.

Agile Development: What Is Agility? What Is an Agile Process? Scrum Other Agile Process Models, Scrum, Other Agile Frameworks- The XP Framework.

UNIT II **CO-2** **14 Periods**

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Requirements gathering, developing use cases, Building the Analysis Model, Negotiating Requirements, Requirements monitoring, Validating Requirements.

Requirements Modelling: Requirements Analysis, Scenario-Based Modeling, Class-Based Modeling, Functional Modelling, Behavioral Modelling.

Design Concepts: Design within the Context of Software Engineering, the Design Process, Design Concepts, the Design Model

UNIT III **CO-3** **14 Periods**

Architectural Design: Software Architecture, Agility and Architecture, Architectural Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Reviews.

Modeling Component-Level Design: What Is a Component? Designing Class-Based Components, Conducting Component Level Design.

User Experience Design: User Experience Design Elements, The Golden Rules, User Interface Analysis and Design, Interface Analysis and Design Models, The process.

UNIT IV **CO-4** **14 Periods**

Software Testing –Component Level: A Strategic Approach to Software Testing, Planning and Record keeping, Test case design, White box testing, Black-Box-Testing.

Software-Testing Integration level: Software Testing Fundamentals, Integration testing, Validation Testing, Testing Patterns.

Software Metrics and Analytics: Software Measurement, Software Analytics, Product Metrics, Metrics for Testing, Metrics for maintenance, Process and Project Metrics, Metrics for Quality.

Learning Resources:

Text Book:

1. Roger Pressman and Bruce Maxim "Software Engineering- A Practitioner's Approach", 9th edition, Tata McGraw-Hill International.

Reference Books:

1. Ian Somerville, Software Engineering. 6 ed, Pearson Education.
2. Carlo Ghezzi, Mehdi Jazayeri and Dino Mandrioli, Fundamentals of Software Engineering 2nd Edition, PHI.
3. RajibMall, Fundamentals of Software Engineering. 2nd Edition , PHI.

Web Resources:

1. http://nptel.ac.in/courses/106101061/2
2. http://nptel.ac.in/courses/106101061/5

CO/CS/IT 313 **DESIGN & ANALYSIS OF ALGORITHMS** **L P C**
3 0 3

Course Objectives:

The main objectives of this course are

1. Instruct performance analysis of an algorithm.
2. Illustrate algorithm design Strategies.
3. Demonstrate pattern matching algorithms
4. Impart knowledge on P, NP and NP-complete and NP-hard class of problems.

Course Outcomes

After completion of the course, the students will be able to

1. Analyze the performance of algorithms.
2. Apply algorithm design techniques to solve real world problem.
3. Make use of string matching algorithms to solve complex problems.
4. Solve P class and NP class problems.

Course Content:

UNIT I **CO1** **10 Periods**

Introduction- What is an Algorithm? Algorithm Specification, Performance Analysis, Randomized Algorithms – Identifying the repeated element, primality testing, advantages and disadvantages.

Divide and Conquer: General Method, Merge Sort, Quick sort, Divide and Conquer Run Time Recurrence Relations.

UNIT II **C02** **15 Periods**

Greedy Programming: General Method, Knapsack problem, Job Sequencing with Dead Lines, Minimum Spanning Tree - Prim's and Kruskal's algorithms, Single Source Shortest-Paths-Dijkstra's.

Dynamic Programming: General Method, Multi Stage Graph, All Pairs Shortest Paths, Single Source Shortest Paths-general Weights, Optimal Binary Search Trees, 0/1 Knapsack, Traveling Salesman Problem.

UNIT III **CO3** **13 Periods**

Back tracking: General Method, 8-queen problem, Hamiltonian Cycles, 0/1 Knapsack.

Branch and Bound: Control Abstraction for LC Search, Bounding, FIFO branch and bound, LC branch and bound, 0/1 Knapsack problem, Travelling Salesman Problem.

UNIT IV **CO4** **12 Periods**

String Matching – The Naïve String-Matching Algorithm, The Rabin-Karp Algorithm, String Matching with Finite Automata, The KMP Algorithm.

NP-Completeness - Polynomial Time, Polynomial Time verification, NP Completeness and reducibility, NP Complete Problems. Approximation Algorithms - The Travelling Sales Persons Problem. (Excluding Theorem Proofs)

Text Book:

1. E. Horowitz, S. Sahni and S.Rajsekaran, "Fundamentals of Computer Algorithms", Galgotia Publication. (Unit I, II, III).
2. T. H. Cormen, Leiserson, Rivest and Stein, "Introduction of Computer Algorithm", PHI. (UnitIV).

Reference Book:

1. Sara Basse, A.V. Gelder, "Computer Algorithms", Addison Wesley.

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

CO351	**Software Computing Lab**	L	T	P	C
		0	0	3	1.5

Course Objectives:
At the end of the course, the student will understand

1. The importance of a component and functionality of each UML model element throughout the software engineering process.
2. How to read and interpret the artifacts of requirements that are used as starting points for analysis and design.
3. Identify and understand interaction among various design model elements.
4. Analyze and design a model or a software component for a particular application or software project

Course Outcomes:
At the end of the course, the student will be able to

1. Describe the importance of systems analysis and design in solving computer Based problems.
2. Develop UML models which are used during the phases of the Rational Unified Process.
3. Illustrate interactions among analysis classes for developing the class model and identify the dynamic behavior of the system.
4. Propose deployment of object-oriented software for client in detail.

LAB CYCLE 01

ANALYSIS

1. Problem Statement
2. Requirements elicitation
3. System Requirements Specification

USECASE VIEW

4. Identification of Actors.
5. Identification of Use cases.
6. Flow of Events.
7. Construction of Use case diagram.
8. Building a Business Process model using UML activity diagram Lab.

LAB CYCLE 02

LOGICAL VIEW

9. Identification of Analysis Classes.
10. Identification of Responsibilities of each class.
11. Construction of Use case realization diagram.
12. Construction of Sequence diagram.
13. Construction of Collaboration diagram.
14. Identification of attributes of each class.
15. Identification of relationships of classes.
16. Analyzing the object behavior by constructing the UML State Chart diagram.
17. Construction of UML static class diagram. Lab Cycle - III

LAB CYCLE 03

DESIGN AND IMPLEMENTATION

18. Refine attributes, methods and relationships among classes.
19. Construction of UML component diagrams.
20. Construction of UML deployment diagrams.

MINI PROJECT

The above three cycles are to be carried out in the context of a problem / information system chosen by the Project batch and a report is to be submitted to the department by the end of the semester.

CO352	IoT with Cloud Computing Lab	L	P	C
		0	3	1.5

Course Objectives:

At the end of this course the students will understand;

1. Overview and various real time usage scenarios of ESP8266 NodeMCU.
2. Various approaches for Controlling devices remotely through cloud.
3. Glimpse of cloud-based sensor data processing and usage.
4. Practical oriented basics of MS Azure, AWS & IBM Watson IoT services.

Course Outcomes:

On completion of this course, students will be able to;

CO1 - Create real-time automation applications with ESP8266 NodeMCU.
CO2 - Controlling sensors through cloud.
CO3 - Process data efficiently through various clous based services.
CO4 - Creating live data instances and visualization of analytics.

List of Experiments:

1. Discuss ESP8266 NodeMCU Pinout with respective Arduino IDE compatibility.
2. Write an Arduino sketch in Arduino IDE to blink on-board LED and externally connected LED of ESP8266 NodeMCU.
3. Write an Arduino sketch to control on-board LED of ESP8266 NodeMCU from the Web Server.
4. Define functionality and pinout of relay and write an Arduino sketch to control AC Bulb using ESP8266 NodeMCU.
5. Write an Arduino sketch to control 2 AC Devices from the Web Server, using ESP8266 NodeMCU.
6. Connect ESP8266 NodeMCU to Arduino IoT Cloud and write an Arduino sketch to read sensor data.
7. Connect ESP8266 NodeMCU to Blynk Cloud platform and write an Arduino sketch to control on-board LED of ESP8266 NodeMCU.
8. Write an Arduino sketch for ESP8266 NodeMCU to monitor temperature and humidity using DHT11 sensor through Blynk cloud platform.
9. Write an Arduino sketch for of ESP8266 NodeMCU to implement Live Water Tank monitoring using Ultrasonic sensor through Blynk cloud platform.
10. Create a Microdoft Azure account and send sensor data from ESP8266NodeMCU to Microsoft Azure IoT Analytics.
11. Setup ESP8266 NodeMCU with IBM Watson IoT Cloud Services and process the sensor data in cloud.
12. Setup ESP8266 NodeMCU with AWS IoT Cloud Services and process the sensor data in cloud.

Professional Elective-I Subjects		
CO 314		
S.NO	**Subject Code**	**Name of the Subject**
1.	CO EL01	IoT with Cloud Computing
2.	CO EL02	Cyber Security
3.	CO EL03	Distributed Systems
4.	CO EL04	Digital Image Processing

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

CO 314 **COEL01: IoT with Cloud Computing** **L P C**

3 0 3

Course Objectives:

1. **LO-1:** To understand the association of Internet of Things with Secure Cloud Computing
2. **LO-2:** To impart knowledge of components of Internet of Things
3. **LO-3:** To understand the potential of famous Cloud Platforms
4. **LO-4:** To develop skills required to build real-life IoT based projects using Cloud Platforms

Course Outcomes:

1. **CO-1:** Identify the need of cloud computing for IoT
2. **CO-2:** Apply Machine Learning techniques for IoT Data
3. **CO-3:** Understanding the Data Analytics with IoT Data
4. **CO-4:** Identify the Vulnerability in Cloud – IoT

Course Content:

UNIT I **CO 1** **12 Periods**

Introduction to Internet of Things(IoT): Concepts and definitions of IoT, IoT data vs big data, IoT Analytics life cycle and Techniques, IoT Complete Technology Chain, Applications of IoT, Opportunities and challenges in IoT.

Cloud Computing: cloud services models, cloud Deployment models, Need of cloud computing for IoT, Fog computing Vs cloud computing for IoT.

UNIT II **CO 2** **12 Periods**

IoT Cloud Platforms: Microsoft Azure IoT, Amazon Webs Services IoT, IBM WATSON IoT, Google's cloud IoT, Principles and foundation of Artificial intelligence and IoT, Machine Learning Paradigms for IoT, Supervised learning for IoT, Linear regression, Logistic regression, SVM-Decision Tree, Naïve's bayes- Deep Learning for IoT

UNIT III **CO 3** **12 Periods**

IoT Analytics for Cloud: Review of IoT analytics for the cloud, IoT Analytics challenges, Microsoft Azure Analytics overview, Designing data processing for analytics, Designing visual analytics for IoT data, Data Science for IoT, Feature engineering with IoT data.

UNIT IV **CO 4** **12 Periods**

IoT with secure Cloud: Cloud Threats in IoT, API's in IoT, Authentication in IoT, Strategies for securing IoT in Cloud, Public key Cryptography.

Text Books:

1. Internet of Things, "A Hands on Approach", by Vijay Madisetti, Arshdeep Bahga, University Press
2. "Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security", by Perry Lea, Packt Publishing Ltd., 2018
3. Internet of Things with Raspberry Pi and Arduino. Boca Raton, by Singh, R., Gehlot, A., Gupta, L., Singh, B., Swain, M, CRC Press, 2020
4. Embedded Systems - SoC, IoT, AI and Real-Time Systems, 4th Edition Kindle Edition by Raj Kamal

References:

1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases, by Pethuru Raj and Anupama C. Raman (CRC Press)

CO 314 **COEL02:CYBER SECURITY** **L P C**
3 0 3

Course Objectives:
The main objectives of this course are:

1. Introduce the fundamental Information security concepts & Threats.
2. Illustrate the security standards and policies to be maintained by the organizations.
3. Describe various Security Performance Metrics & Configuration reviews.
4. Discuss the different log management and backup procedures.

Course Outcomes:
After successful completion of the course, the students are able to

1. Analyze the Information Security Assets and Threats.
2. Identify the various security standards and policies to be maintained by the organizations.
3. Design and Implement Security Performance Metrics, Configuration reviews, and log management.
4. Apply the backup procedures, and Security Audit process using Vulnerability analysis tools.

UNIT I **CO 1** **13 Periods**

Information Security Assets & Threats: Introduction, Role of a security analyst, Threats, Virus, Worms, Trojans, Other Threats, types of Network Attacks, types of Phishing Attack, Types of viruses, Types of worms, types of Trojans. DoS (denial-of-service) attack, Common Vulnerabilities and Exposures(CVE), Bluetooth related attacks.
Fundamentals of Information Security: Elements of information security, Principles and concepts - data security, Types of controls, Discretionary Access Control (DAC), Role- Based Access Control(RBAC).

UNIT II **CO 2** **13 Periods**

Roles and Responsibilities: Information and Data Security Team, CEO or Executive Management, Security Engineer, Systems Administrator, Security Steering Committee, Security Incident Response Team.
Data Leakage: Introduction – Data Leakage, Organizational Data Classification, Location and Pathways, Content Awareness, Content Analysis Techniques, Data Protection, DLP Limitations, DRM-DLP Conundrum, Case studies: SQL Injection using OWASP tool. Information Security Policies, Procedures.
Standards and Guidelines: Information Security Policies, Key Elements of a Security Policy, Security Policy implementation, Security Standards, COSO, COBIT, ISO 27001, SANS.

UNIT III **CO 3** **13 Periods**

Information Security Performance Metrics: Introduction–Security Metrics, Types of Security Metrics, Using Security Metrics, Developing the Metrics Process, Metrics and Reporting.
Configuration review: Configuration Management, Organizational Sec CM Policy, Identify CM Tools, Implementing Secure Configurations, case studies.
Log Correlation and Management: Event Log Concepts, Log Management Infrastructure and functions, Log Management-Using Log watch.

UNIT IV **CO 4** **12 Periods**

Data Backup: Types of Backup, Backup Procedures, Types of Storage, Features of a Good Backup Strategy. Information Security Audit: Information Systems Audit versus Information Security Audit, What is an Information Security Audit, Scope of the Audit, Types of Security Audits, Phases of Information Security Audit, Information Security Audit Methodology, Role of an Auditor, Penetration testing stages.

Vulnerability Analysis: What Is Vulnerability Assessment, Vulnerability Classification, Types of Vulnerability Assessment, Vulnerability Analysis Tools, Case studies.

Text Book:

1. NASSCOM Handbook Study Material.

Reference Books:

1. Nina God bole, “Information System Security”, Wiley
2. Bothra Harsh, “Hacking”, Khanna Publishing House, Delhi.
3. GeorgeK. Kostopoulous, Cyber Space and Cyber Security, CRC Press, 2013.
4. MarttiLehto, Pekka Neittaanmäki, Cyber Security: Analytics, Technology andAutomation edited, Springer International Publishing Switzerland 2015
5. Nelson Phillips and EnfingerSteuart, ― Computer Forensics and Investigations &# 8214; Cengage Learning NewDelhi,2009.

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

CO 314 **COEL03: DISTRIBUTED SYSTEMS** **L P C**
3 0 3

Course Objectives

The main objectives of this course are:

1. Introduce basics in distributed systems and their architectures.
2. Discuss multi- threading concepts and naming in distributed systems.
3. Illustrate synchronization among distributed applications, consistency protocols and Replica management in distributed file systems.
4. Discuss paradigms used to organized is tribute systems.

Course Outcomes

After completion of this course, Students will able to

1. Describe the basic issues, architectures and communication mechanisms in distributed systems.
2. Define processes and naming concepts in distributed systems.
3. Describe synchronization, consistency and replication of distributed applications.
4. Define Fault Tolerance and distributed file systems concepts.

UNIT I **CO 1** **13 Periods**

Introduction: Definition of a Distributed System, Goals, types of distributed systems. Architectures: Architectural Styles, System Architectures, Architectures Versus Middleware. Communication: Fundamentals, Remote Procedure Call, Message-Oriented Communication, Stream-Oriented Communication, Multicast Communication.

UNIT II **CO 2** **13 Periods**

Processes: Threads, Virtualization, Clients, Servers, Code Migration.
Naming: Names, Identifiers, and Addresses; Flat Naming, Structured Naming, Attribute-Based Naming.

UNIT III **CO 3** **12 Periods**

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

UNIT IV **CO 4** **12 Periods**

Fault Tolerance: Introduction To Fault Tolerance, Process Resilience, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit, Recovery.
Distributed File Systems: Architecture, Processes, Communication, Naming, Synchronization, Consistency and Replication, Fault Tolerance.

Text Book:

1. Andrew S.Tanenbaum, Maarten Van Steen, Distributed Systems: Principles and Paradigms, 2nd Edition, Pearson Education/PHI.

Reference Books:

1. George Coulour is, Jean Dolli more, TimKind berg, Distributed Systems Concepts and Design 3rd edition, Pearson Education.
2. Mukesh Singhal & Niranjan G.Shivaratri, Advanced Concepts in Operating Systems, Tata Mc.Graw Hill edition 2001.
3. Pradeep Kumar Sinha, Distributed Operating System-Concepts and Design, PHI.

WEBREFERENCES:

1. www.cis.upenn.edu/~lee/00cse380/lectures/
2. www.cs.uah.edu/~weisskop/Notes690/

CO 314 **COEL04: Digital Image Processing** **L P C**

3 0 3

Course objectives:

The main objectives of this course are to:

1. To create basic understanding of fundamental concepts in digital image processing and enhancement in the spatial domain.
2. To demonstrate the approaches used in enhancement in the frequency domain and image segmentation.
3. Tot each image restoration and image compression techniques.
4. To analyze morphological transformations, and image representation of real world objects

Course outcomes:

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

1. Define image processing systems and develop algorithms for image enhancement techniques in the spatial domain.
2. Implement enhancement techniques in the frequency domain and image segmentation.
3. Develop image restoration, and image compression techniques.
4. Analyze morphological transformation algorithms, and select various descriptors for image representation.

Course Content:

UNITI **CO1** **12 Periods**

Introduction: Digital Image Processing, Fundamental Steps in Digital Image Processing, Components of an Image Processing System.

Digital Image Fundamentals: Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Some basic Relationships between Pixels.

Image Enhancement in the Spatial Domain: Some Basic Gray Level Transformation, Histogram Processing, Enhancement Using Arithmetic / Logic Operations, Basics of Spatial Filtering, Smoothing spatial Filters, Sharpening spatial Filters.

UNIT II **CO2** **12 Periods**

Image Enhancement in the Frequency Domain: Introduction to the Fourier Transform and the Frequency Domain, Smoothing frequency domain Filters, Sharpening frequency-domain Filters, Holomorphic Filtering, Implementation.

Image Segmentation: Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation.

UNIT III **CO3** **12 Periods**

Image Restoration: A Model of the Image Degradation/Restoration Process, Linear, Position-In variant Degradations, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Squares Filtering.

Image Compression: Image Compression Models, Error-free Compression, Lossy Compression, Image Compression Standards.

UNIT IV **CO4** **12 Periods**

Morphological Image Processing: Dilation and Erosion, The Hit-or-Miss Transformation, Some basic Morphological Algorithms, Extension to Gray-Scale Images.

Representation and Description: Representation, Boundary Descriptors, Regional Descriptors.

Learning Resources:

Text Books:

1. Rafael C.Gonzalez, Richard E.Woods, 'Digital Image Processing 'Addison Wesley Pubs (2nd Edition).

Reference Books:

1. "Image Processing. Analysis, and Machine Vision", MilanSonka, VaclavHlavac, Roger Boyle (2nd Edition).
2. A.K.Jain, 'Fundamentals of Digital Image Processing' PHI.

Job Oriented Elective-I Subjects		
CO 315		
S.NO	**Subject Code**	**Name of the Subject**
1.	JOEL01	Artificial Intelligence
2.	JOEL02	FOG Computing

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

CO 315 **JOEL01: ARTIFICIAL INTILIGENCE** **L P C**
3 0 3

Course Objectives:

The main objectives of this course are to:

1. Introduce fundamental concepts of artificial intelligence.
2. Impart knowledge on problem solving using uninformed, informed, local and adversarial search strategies.
3. Create awareness on formalization of knowledge and reasoning.

Course Outcomes:

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

1. Explain the fundamental concepts of artificial intelligence
2. Apply search strategies for solving real world problems
3. Utilize game playing strategies for solving problems
4. Infer knowledge using propositional and predicate logic
5. Discuss knowledge representation of the real world using On to logies
6. Summarize the algorithms for classical planning

Course Contents:

UNIT I **CO1** **10 Periods**

Introduction to AI: What Is AI? The Foundations of AI, The History of AI, The State of the Art.
Intelligent Agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents.
Problem Solving by Search: Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.

UNIT II **CO2** **14 Periods**

Beyond Classical Search: Local Search Algorithms and Optimization Problems, Searching with Non-Deterministic Actions.
Adversarial Search: Games, Optimal Decisions in Games, Alpha–Beta Pruning.
Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Constraint Propagation, Back tracking Search for CSPs, Local Search for CSPs, The Structure of Problems.

UNIT III **CO3** **12 Periods**

Logical Agents: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Effective Propositional Model Checking, Agents Based on Propositional Logic.
First-Order Logic: Representation Revisited, Syntax and Semantics of First-Order Logic, Using First Order Logic, Knowledge Engineering in First-Order Logic.
Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

UNIT IV **CO4** **14Periods**

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

Automated Planning: Definition of Classical Planning, Algorithms for Classical Planning and Acting in the Real World: Time, Schedules and Resources, Hierarchical Planning.

Learning Resources:

Text Books:

1. Artificial Intelligence-A modern Approach, Stuart Russell and Peter Norvig, Fourth Edition, Pearson Education

References:

1. Artificial Intelligence, E.Rich and K.Knight,3rd Edition.,(TMH)
2. Artificial Intelligence, 3rd Edition., Patrick Henny Winston, 3rd Edition., Pearson Education.
3. A First Course in Artificial Intelligence, Deepak Khemani, TataMc-GrahHill.
4. Artificial Intelligence and Expert systems–Patterson, Pearson Education.
5. Artificial Intelligence, Saroj Kaushik, CENGAGE Learning.

CO 315 **JEL02: FOG COMPUTING** **L P C**
3 0 3

Course Objectives:
At the end of this course the students will understand
1. To introduce the structure and functions of the Fog Computing
2. To provide the knowledge of how the Fog Computing works hand in hand with IoT
3. To expose the students to the issues related to Networking and Security issues in the Fog Computing systems.

Course Outcomes:
1. Describe the fundamental concepts of a Fog Computing functionality, and architecture.
2. Identification and Application of Fog Computing concepts in real-time daily life application scenarios.
3. Analyse how the performance of IoT increases with Fog Computing interoperability.

Course Contents:

UNIT I **CO1** **12 Periods**
Introduction to Fog Computing: Fog Computing, Characteristics, Application Scenarios, Issues and challenges.
Fog Computing Architecture: Communication and Network Model, Programming Models, Fog Architecture for smart cities, healthcare and vehicles.
Fog Computing Communication Technologies: Introduction, IEEE 802.11, 4G and 5G standards, WPAN, LPWAN

UNIT 2 **CO2** **10 Periods**
Fog Computing Requirements when applied to IoT: Scalability, Interoperability, Fog-IoT architectural model, Challenges on IoT Stack Model via TCP/IP Architecture, Cloudification, Virtualization, Security and Privacy issues.

UNIT 3 **CO3** **11 Periods**
Software Defined Networking and application in Fog Computing: Open Flow Protocol, Open Flow Switch, SDN in Fog Computing, Home Network using SDN.
Security and Privacy issues: Trust and privacy issues in IoT Network, Web Semantics and trust Management for Fog Computing, Machine Learning based security in Fog Computing, Cyber Physical Energy Systems over Fog Computing.

UNIT 4 **CO4** **12 Periods**
Exploiting Fog Computing in Health Monitoring: An Architecture of a Health Monitoring IoT Based System with Fog Computing, Fog Computing Services in Smart E Health Gateways.
Fog Computing Model for Evolving Smart Transportation Applications: Introduction, Data-Driven Intelligent Transportation Systems, Fog Computing for Smart Transportation Applications.

Text Books:

1. Fog Computing: Theory and Practice by Assad Abbas, Samee U. Khan, Albert Y. Zomaya
2. Fog and Edge Computing: Principles and Paradigms (Wiley Series on Parallel and Distributed Computing) by Rajkumar Buyya and Satish Narayana Srirama

SKILL Oriented Courses III-Subjects		
COSL3		
S.NO	**Subject Code**	**Name of the Subject**
1.	a	Web Design and Development
2.	b	2D-Computer Animation
3.	c	Unix Programming

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

COSL301	Web Design & development	L P C
		1 2 2

Course Objectives:

1. To enable students to develop modern web application by leveraging latest technologies
2. To build strong foundation in students making them job ready as per industry requirements
3. To enable them to learn new technologies by applying foundation paradigms
4. To building strong expertise to develop end to end application - web frontend and backend development

Course Outcomes:

After completion of the course, the student should be able to:

CO-1: Build static and dynamic web pages with HTML, XML, JSON CO-
2: Create Dynamic web pages using CSS and Java Script
CO-3: Understand the concepts, analyze and build interactive web applications
CO-4: Apply various frameworks of web technologies to optimize the applications

Course Content:

UNIT I **CO 1** **12 Periods**

Introduction: Concept of website, its need and purpose, Types of websites: Static and dynamic website, Introduction to HTML, XML, JSON, Web Browsers, – Web Servers, Uniform Resource Locator, Tools and Web Programming Languages. Web Standards,

Tiered Architecture: Client Server Model, Three Tier Model, Service Oriented Architectures, REST services.

UNIT II **CO 2** **12 Periods**

Hypertext Mark Up Language: Languages used for website development, HTML5: basic tags, formatting tags, Adding images, Lists, Embedding multimedia in Web pages, Inserting tables, Internal and External Linking, Frames, Forms

UNIT III **CO 3** **10 Periods**

Cascading Style Sheets (CSS3): Basics of Cascading Style sheets, Advantages of CSS, External Style sheet, Internal style sheet, Inline style sheet, CSS Syntax, color, background, Font, images.

Java Script: Features of JavaScript, extension of JavaScript, Syntax of JavaScript: data types, operators, variables, tag, Document Object Model (DOM) with JavaScript, Selection Statement using if and Switch, Iterative statement: for, for/in, while, do while, break and continue

UNIT – IV **CO 4** **11 Periods**

Front End Framework: Introduction to jQuery - Syntax, Selectors, Events, Traversing, AJAX; Introduction toBootstrap – Basics, Grids, Themes; Angular JS – Expressions, Modules, Data Binding, Scopes, Directives & Events, Controllers, Filters, Services, Validation

Back End Technologies: Introduction to Restful services, Resources, Messages (Request, Response), Addressing, Methods – (GET, POST, PUT, DELETE)

Text Books:

1. Internet and World Wide Web: How to Program, Deitel P. J., Deitel H. M. and Deitel5th Edition, Pearson Prentice Hall, 2012
2. HTML & CSS: Design and Build Websites, Jon Duckett, John Wiley & Sons

Reference Books:

1. Programming the World Wide Web, Sebesta R. W, 8th edition, Pearson, 2014
2. Web Engineering: a practitioner's approach, Pressman R. and Lowe D,1st Edition, McGrawHill,2008
3. Web Engineering: The Discipline of systematic Development of Web Applications, Kappel G., et al, 1st Edition, John Wiley & Sons, 2006
4. Web Engineering: Principles and Techniques, SuhW, Idea Group Inc, 2005
5. PHP for the Web: Visual Quick Start Guide, Ullman L, 5th Edition, Peachpit Press, 2016

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

COSL302 **2-D Computer Animation** **L P C**
1 2 2

Course Objectives:

1. Developing the basic skills necessary for the student to produce digital character-based animation, titles for film and video.
2. Learning and experiencing the arts of storytelling, animation and cinematography while making 2D animation movies, motion graphics, and GIF stickers
3. Understanding principles that translate sequential images into action to make animation Believable

Course Outcomes:

1. Define and apply design principles and theories to animation production.
2. Evaluate and apply the 12 principles of animation based on the requirements of the storyline.
3. Demonstrate progress in basic drawing and animation skills
4. Create traditional and computer generated 2D animation based on current industry trends and Practices

COURSE CONTENT:

UNIT I **CO1** **10 Periods**

History of animation: Types of animation, Understanding and learning the Principles of animation through the view of different animation films.
Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

UNIT II **CO2** **8 Periods**

Basic transformation: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms transformations between coordinate systems

UNIT III **CO3** **10 Periods**

2-D Viewing: introduction to 2-D Viewing, The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen Sutherland and Cyrus-beck
line clipping algorithms, Sutherland –Hodge man polygon clipping algorithm.

UNIT IV **CO4** **12 Periods**

2D Software Interface: Understanding the 2d software interface Drawing tools, pen tools and other necessary tools to create any drawing in the frames.
Frame by frame animation: Creating frame by frame animation for a short animation (maximum 10 sec with simple drawing).
Tween: Creating simple animation with shape, classic & motion tweening. Creating simple animation with shape and classic tween together.

TEXT BOOKS:

1. Frank Thomas and Odie Johnson, The Illusion of Life: Disney Animation, Disney Editions; Rev Sub edition, 2014
2. Neuman and Sproul, Principles of Interactive Computer Graphics, Tata McGraw Hill, 2nd edition., 1978

REFERENCES:

1. Williams, R. The Animator's Survival Kit. Revised Edition, Faber & Faber, 2011
2. Donald Hearn and M. Pauline Baker, Computer Graphics C version, Pearson education, Second Edition, 2008

R V R & J C COLLEGE OF ENGINEERING (Autonomous), CHOWDAVARAM, GUNTUR-19
B.Tech., Computer Science & Engineering (IoT)

COSL303 **UNIX Programming** **L P C**
1 2 2

Course Objectives:

The objectives of the course are:

1. Introduce UNIX Architecture and its key features.
2. Impart UNIX commands and AWK programming
3. Discuss functions of UNIX shells and the concepts of Bourn shell programming.
4. Demonstrate file and process management system calls signal handling mechanism in UNIX.
5. Demonstrate signal handling mechanism and IPC mechanisms like pipes, shared memory, and semaphores in UNIX

Course Outcomes:

After successful completion of the course, the students are able to

1. Apply UNIX commands for solving problems and work with AWK programming.
2. Develop shell scripts for solving problems that can't be solved by simple commands.
3. Apply system calls for system programming.
4. create applications using basic IPC mechanisms like pipes, shared memory, and semaphores.

Course Content:

UNIT I **CO 1** **10 Periods**

Introduction: UNIX architecture, Features of UNIX.

UNIX Utilities: pwd, mkdir, ls, cd, rmdir, cat, more, page, head, tail, editing a file: vi, cp, mv,rm, wc, ln, unlink, chmod, chown, chgrp, who, sort, nl, grep, egrep, fgrep, find, cmp, diff, uniq,tr, sed,cut,paste,join,tee,tty.

Programmable text processing: AWK - awk programs, accessing individual fields, Begin and end, operators, variables, control structures, extended regular expressions, condition ranges, field separators, Built-in functions.

UNIT II **CO 2** **10 Periods**

UNIX Shells: Introduction, shell functionality, Built-in commands ,meta characters, Input/output redirection, filename substitution, pipes, command substitution, sequences, grouping commands, background processing, scripts, subshells, shell variables, Quoting

Bourne Shell: Working with variables, Arithmetic, conditional expressions, control structures, positional parameters, passing command line arguments, shell programs, functions, and arrays.

UNIT III **CO 3** **10 Periods**

File Management: Introduction to system calls and file management, Regular file management system calls - open(), read(), write(), lseek(), Close(), unlink(), stat(), getdents(). Miscellaneous file management system calls – chown() and fchown(), chmod () and fchmod (), dup () and dup2(), fcntl(), ioctl(), link(), mknod(), sync(), truncate() and ftruncate().

Process Management: Creating a new process –fork (), orphan processes, terminating a process – exit (), zombie processes, waiting for a child – wait (), Differentiating a process – exec(),changing directories– chdir(), changing priorities- nice(), Accessing user and Group ID's.

UNIT IV **CO 4** **10 Periods**

Signals: Introduction, A list of signals, terminal signals, Requesting an Alarm signal - alarm (),handling signals - signal (), protecting critical code and chaining interrupt handlers, sending signals - kill (), Death of children, suspending and Resuming processes, process Group's and control terminals.

Inter process communication: Pipes, shared memory and semaphores.

Learning Resources:

Text Book:

1. "Unix for programmers and users "3rd edition by Graham Glass, King Ables, Pearson education.

Reference Book(S):

1. Behrouz A. Forouzan, Richard F. Gilberg: UNIX and Shell Programming- Cengage Learning– India Edition. 2009.
2. W. Richard Stevens, Advanced programming in the Unix environment, 3rd Edition Pearson education.
3. Kernighan W.Brian and PikeRob, Unix programming environment, Pearson education.
4. Sumitabha Das, Your Unix the ultimate guide, TMH2rdedition.
5. Marc J.Rochkind, Advanced UNIX programming, 2nd edition Pearson Education.
6. Meeta Gandhi, Rajiv Shah, Tilak Shetty, The "C" Odyssey UNIX - The Open, Boundless C, BPB Publications.

Web Resources:

1. www.webreference.com>Programming
2. www.iu.hio.no/~mark/unix/unix.html

Web References:

1. Unix Basics(http://www.tutorialspoint.com/unix/)
2. Bourn Shell(http://www.shellscript.sh/)
3. System calls(http://www.tutorialspoint.com/unix_system_calls/)