DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Computer Networks

3150401

COURSE OBJECTIVES

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

Unit-I

Introduction: Computer Network, Types- LAN, MAN & WAN,Data transmission modes- Serial & Parallel, Simplex, Half duplex & full duplex, Synchronous & Asynchronous transmission, Transmission medium- Guided & Unguided, Cables- Twisted pair, Coaxial cable & Optical fiber, Networking devices-Repeaters, Hub, Switch, Bridge, Router, Gateway, Modem, Proxy Server, Wireless router, & Wireless Access Point (WAPs). Performance Criteria- Bandwidth, Throughput, Latency (Delay), Propagation Time, Transmission time & Queuing Time, Network Standardization- OSI Reference Model & TCP/IP Reference Mode.

Unit-II

Physical Layer: Network topologies- Bus, Ring, Star Topology & Mesh, Switching- Circuit switching, Message switching & Packet switching, Multiplexing; FDM – Frequency division multiplexing, WDM – Wavelength division multiplexing & TDM – Time division multiplexing, Wireless transmission- Electromagnetic spectrum, Radio transmission & Microwave transmission.

Unit-III

Data Link Layer: Introduction, Design issues, Services, Framing, Error control, Flow control, ARQ Strategies, Error Detection and correction, Parity bits, Cyclic Redundant Code (CRC), Hamming codes, MAC Sub Layer- The channel allocation problem, Pure ALOHA ,Slotted ALOHA, CSMA ,CSMA/CD,CSMA/CA,IEEE 802.3 frame format.

Unit-IV

Network Layer& Transport Layer: Introduction, Design issues, Services, Routing- Distance vector routing, Hierarchical routing , Link state routing, Shortest path algorithm- Dijkstra's Algorithm & Floyd–Warshall's Algorithm, Flooding, Congestion Control- Open

Loop & Closed Loop Congestion Control, Leaky Bucket & Token bucket Algorithm. Connection Oriented & Connectionless Service, Port addressing basics

Unit-V

Presentation, Session& Application Layer: Introduction, Design issues, Presentation layer- Translation, Encryption & Compression. Session Layer – Dialog Control, Synchronization. Application Layer- Remote login, File transfer & Electronic mail.

Network Function Virtualization (NFV) - Architecture and Concepts, Programmable Networks - Introduction to P4, SmartNICS and P4 switches, Data Center Networking (DCN) – Introduction, DCN - Deep Dive (Network topologies, Container Network Interfaces)

.

RECOMMENDED BOOKS

- Behrouz A. Forouzan "Data Communication and Networking", McGraw – Hill Publications.
- Andrew Tanenbaum – Computer Networks, PHI
- Peterson and Davie, "Computer Networks, A systems Approach", 5th ed., Elsevier, 2011.
- Ying-Dar Liu, Ren-Hwang, Fred Baker, "Computer Networks: An open Source Approach", McGraw – Hill, 2001.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Outline the Data Communications System and its components.

CO2. Identify the different types of network topologies and protocols.

CO3. Enumerate the layers of the OSI model and function(s) of each layer.

CO4. Identify the different types of network devices and their functions within a network

CO5. Analyze the problems associated with various networking protocols and measure the Performance

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	3								2	2			
CO2	3	2	3	3		3		2			3	2	3	
CO3	3	3	3		1						2	3	3	
CO4				2	2	3	1	1	1	1	1		1	
CO5				2	2	3	1	1	1	1	1	1	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Computer Networks

3150401

List of Experiments

1. Basic Network Configuration

- Setting up a local area network (LAN) using routers and switches.
- Configuring static IP addresses and testing connectivity using **ping** and **tracert/traceroute**.
- Verifying network connectivity with tools like **netstat**, **arp**, and **ipconfig/ifconfig**.

2. Protocol Implementation and Analysis

- **TCP/UDP Socket Programming**: Writing client-server programs for data exchange.
- Understanding and simulating the **handshake process in TCP**.
- Packet sniffing and analysis using tools like **Wireshark**.

3. Network Simulation

- Simulating networks with tools like **Cisco Packet Tracer**, **NS3**, or **GNS3**.
- Setting up and analyzing **routing protocols** (e.g., RIP, OSPF, EIGRP).

4. Routing and Switching

- Configuring routers and switches for VLANs.
- Setting up and verifying **static routing** and **dynamic routing protocols**.
- Troubleshooting network connectivity issues with routing tables.

5. Application Layer Experiments

- Setting up web servers and file servers (e.g., **Apache**, **NGINX**, or **FTP servers**).
- Configuring and testing **DNS** and **DHCP** servers.

6. Network Security

- Implementing **firewalls** and configuring access control lists (ACLs).
- Setting up secure connections using **VPNs**.
- Analyzing and mitigating common network attacks (e.g., ARP spoofing, DDoS attacks).

7. Network Performance Analysis

- Measuring **latency**, **throughput**, and **packet loss** using tools like **iperf**.
- Monitoring bandwidth usage and network performance using tools like **Nagios**, **Zabbix**, or **SolarWinds**.

Computer Networks

3150401

List of Skill Based Mini Project

1. Develop a simple client-server chat system using TCP or UDP.
2. Create a tool that mimics the working of the ping command, including ICMP packet creation and response handling.
3. Build a tool to calculate subnets, including network IDs, broadcast addresses, and usable host ranges.
4. Create an application that resolves domain names to IP addresses without using the system DNS.
5. Build a basic HTTP server to serve static web pages.
6. Develop a file-sharing system using TCP sockets.
7. Implement a packet-capturing application using Python's scapy or similar libraries.
8. Create an application to measure network bandwidth and latency.
9. Build a tool to monitor active connections, traffic usage, or specific port activity.
10. Design a basic firewall to block specific IPs or ports using Python or Linux iptables.
11. Build a basic brute force or dictionary attack tool for ethical hacking purposes (on test cases).
12. Create a tool that detects ARP spoofing attacks in a network.
13. Develop a system to stream video over UDP or TCP.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Computer Networks

2150411(OLD)

COURSE OBJECTIVES

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

Unit-I

Introduction: Computer Network, Types- LAN, MAN & WAN,Data transmission modes- Serial & Parallel, Simplex, Half duplex & full duplex, Synchronous & Asynchronous transmission, Transmission medium- Guided & Unguided, Cables- Twisted pair, Coaxial cable & Optical fiber, Networking devices-Repeaters, Hub, Switch, Bridge, Router, Gateway, Modem, Proxy Server, Wireless router, & Wireless Access Point (WAPs). Performance Criteria- Bandwidth, Throughput, Latency (Delay), Propagation Time, Transmission time & Queuing Time, Network Standardization- OSI Reference Model & TCP/IP Reference Mode.

Unit-II

Physical Layer: Network topologies- Bus, Ring, Star Topology & Mesh, Switching- Circuit switching, Message switching & Packet switching, Multiplexing; FDM – Frequency division multiplexing, WDM – Wavelength division multiplexing & TDM – Time division multiplexing, Wireless transmission- Electromagnetic spectrum, Radio transmission & Microwave transmission.

Unit-III

Data Link Layer: Introduction, Design issues, Services, Framing, Error control, Flow control, ARQ Strategies, Error Detection and correction, Parity bits, Cyclic Redundant Code (CRC), Hamming codes, MAC Sub Layer- The channel allocation problem, Pure ALOHA ,Slotted ALOHA, CSMA ,CSMA/CD,CSMA/CA,IEEE 802.3 frame format.

Unit-IV

Network Layer& Transport Layer: Introduction, Design issues, Services, Routing- Distance vector routing, Hierarchical routing , Link state routing, Shortest path algorithm- Dijkstra's Algorithm & Floyd–Warshall's Algorithm, Flooding, Congestion Control- Open Loop & Closed Loop Congestion Control, Leaky Bucket & Token bucket Algorithm. Connection Oriented & Connectionless Service, Port addressing basics

Unit-V

Presentation, Session& Application Layer: Introduction, Design issues, Presentation layer- Translation, Encryption & Compression. Session Layer – Dialog Control, Synchronization. Application Layer- Remote login, File transfer & Electronic mail.

RECOMMENDED BOOKS

- Behrouz A. Forouzan "Data Communication and Networking", McGraw – Hill Publications.
- Andrew Tanenbaum – Computer Networks, PHI
- Peterson and Davie, "Computer Networks, A systems Approach", 5th ed., Elsevier, 2011.
- Ying-Dar Liu, Ren-Hwang, Fred Baker, "Computer Networks: An open Source Approach", McGraw – Hill, 2001.

COURSE OUTCOMES

Aftercompletionofthecoursestudentswould beable to:

CO1. Outline the Data Communications System and its components.

CO2. Identify the different types of network topologies and protocols.

CO3. Enumerate the layers of the OSI model and function(s) of each layer.

CO4. Identify the different types of network devices and their functions within a network

CO5. Analyze the problems associated with various networking protocols and measure the Performance

CO6. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation

---

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Software Engineering

3150402

COURSE OBJECTIVES

- To understand fundamental concepts of software engineering, including software development life cycles, methodologies, and processes.
- To Develop the ability to systematically gather, analyze, and document software requirements.
- To gain proficiency in software design methodologies and principles, enabling the creation of robust, scalable, and maintainable software architectures.
- To learn and apply various software testing methodologies to ensure software quality and reliability.

Unit-I

Introduction to Software Engineering: Definition, software engineering-layered Technology, Software Characteristics and Components, Software model: Software Development of Life Cycle Model (SDLC), The Waterfall Model, Iterative Waterfall Model, Prototyping Model, Spiral Model, RAD Model, Selection criteria of model: Characteristics of Requirements, Status of Development Team, Users participation, Type of Project and Associated Risk.

Unit - II

Requirement Engineering: Definition, Requirement Engineering Activity , Types of Requirement- Functional and Non-functional Requirements, User and System Requirements, Requirement Elicitation Methods, Requirement Analysis Methods, Requirement Documentation (SRS), Requirement Validation, Requirement Management.

Unit – III

Design Concept, Principle and Methods: Design Fundamentals, Design Principles, Effective Modular Design, Design Representations, Architectural design, Procedural

design, data Directed design, Real Time Design, Object Oriented Design, Coupling and Cohesion.

Unit – IV

Software Metrics, Project Management and Estimation: Metrics in Process and Project domains, Software Measurement, Software Quality Metrics, Project Management- Basics- People, Product, Process, Project, Estimation- Software Project Estimation, Decomposition Techniques- Function Point Estimation, Line of Code (LOC) based estimation, Empirical Estimation, COCOMO Model, Project Scheduling Techniques.

Unit – V

Software Testing: Definitions, Software Testing Life Cycle (STLC), Test Case Design, Strategic Approach to Software Testing- Verification & Validation , Strategic issues, Criteria for completion of Testing, Unit Testing, Integration Testing, Validation Testing, System Testing, Black Box Testing Techniques, White Box Testing Techniques, Acceptance Testing, introduction to security aspects and ethical AI in software engineering, Sustainable Software Lifecycle Management.

RECOMMENDED BOOKS

- Software Engineering, Sommerville, Pearson.
- Software Engineering: A Practitioner's Approach, Roger S. Pressman, McGraw Hill.
- Software Engineering, K.K. Agrawal & Yogesh Singh, New Age Publication.
- Software Engineering, Rajib Mall, PHI.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. **Explain** the various fundamental concepts of software engineering.
CO2. **Recognize** the importance of requirements engineering in the software development lifecycle
CO3. **Identify Effective Software Design principles including cohesion and coupling**
CO4. **Implement** software metrics for estimating the cost, effort, and schedule of software projects.
CO5. **Examine** various testing techniques based on software requirements and design specifications.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	2	2	1	1	1	1	1	1	1	3	-
CO2	2	3	3	3	2	1	1	1	1	1	1	1	3	3
CO3	2	2	3	3	2	1	1	1	1	1	1	2	3	3
CO4	3	3	2	2	3	1	1	1	1	1	1	2	3	3
CO5	3	2	2	2	3	1	1	1	1	1	1	3	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

DATA MINING & WAREHOUSING

3150403

COURSE OBJECTIVES:

- To understand the value of data mining in solving real-world problems.
- To gain understanding of algorithms commonly used in data mining tools.
- To develop ability for applying data mining tools to real-world problems

UNIT-1:

Introduction: Motivation: Important, Data type for Data Mining: Relational Databases Data Ware-Houses. Transactional Databases, Advanced Database System and Its Applications, Data Mining Functionalities Concept/Class Description, Association Analysis Classification & Prediction, Cluster Analysis, Outliner Analysis Classification of Data Mining Systems, Major Issues in Data Mining

UNIT-2:

Data Warehouse and OLTP Technology for Data Mining: Differences between Operational Database Systems & Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Data Cube Technology, Emerging Scenario of Pattern Warehousing System

UNIT -3:

Data Pre-processing: Data Cleaning, Data Integration and Transformation, Data Reduction Discretization and Concept Hierarchy Generation. Data Mining Primitives Languages and System Architectures, Concept Description, Characterization and Comparison Analytical Characterization.

UNIT-4:

Mining Association Rules in Large Databases: Association Rule Mining Market Basket Analysis, Basic Concepts, Mining Single Dimensional Boolean Association Rules from Transactional Databases: The Apriori Algorithm, Generating Association Rules from Frequent Items, Improving the Efficiency of Apriori, other Algorithms & their Comparison, Mining Multilevel Association Rules, Multidimensional Association Rules, Constraint Based Association Rule Mining.

UNIT -5:

Social Media Mining: Social Network Analysis, Influencer Detection, Data Warehousing in Cloud: Google BigQuery, Amazon Redshift, Microsoft Azure Synapse, Data Lake Concepts: Data Lake vs Data Warehouse, Tools (Hadoop, Databricks), Integration with AI and ML: Predictive Analytics,

Real-time Analytics using warehousing tools, Project-Based Learning: Mini projects in applications like customer segmentation and recommendation systems etc.

RECOMMENDED BOOKS

• Data Mining: Concepts and Techniques, Han and Kamber, Morgan Kaufmann Publications.

• Data Mining Techniques, A. K. Pujari, Universities Press Pvt. Ltd .

COURSE OUTCOMES:

After completion of this course, the students would be able to:

CO 1. **Understand** the Fundamentals of Data Mining.

CO2. **Apply** data mining techniques to extract actionable insights and patterns..

CO3**. Understand** and Design Data Warehouse Architectures.

CO4. **Apply** the Apriori Algorithm for Single-Dimensional Rule Mining.

CO5. **Utilize** Cloud Data Warehousing Tools.

CO-PO mapping

Course Articulation Matrix

1 -

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3		3								3	3	3
CO2	3	3	2	3	3		3		2			3	2	3
CO3	3	3	3	3								2	3	3
CO4	3	3	3	3						2		3	2	2
CO5	3	3	3	3	3				3			3	3	3

Slightly; 2 - Moderately; 3 – Substantially

DATA MINING & WAREHOUSING

3150403

List of Experiment

1. Installation of WEKA Tool
2. Creating new Arff File
3. Data Processing Techniques on Data set
4. Data cube construction – OLAP operations
5. Implementation of Apriori algorithm
6. Implementation of FP- Growth algorithm
7. Implementation of Decision Tree Induction
8. Calculating Information gains measures
9. Classification of data using Bayesian approach
10. Implementation of K-means algorithm
11. Build Data Warehouse and Explore WEKA
12. Case study of open source data mining tools (WEKA, ORANGE & TERADATA)

DATA MINING & WAREHOUSING

3150403

List of Skill Based Mini Project

1. Develop a housing price predictions
2. Develop a smart health disease prediction using Naive Bayes
3. Develop a handwritten digit recognition system
4. Develop a movie recommendation system
5. Develop a breast cancer detection
6. Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.,)
7. Develop a Credit Risk Assessment system
8. Develop a sales forecasting using Walmart dataset
9. Develop a D-mart sales prediction
10. Develop a Music recommendation system
11. Develop a system for detecting suicidal tendency
12. Develop a weather forecasting using data mining
13. Develop a customer behaviour prediction using web usage mining

DATA MINING & WAREHOUSING

2150413(OLD)

COURSE OBJECTIVES:

- ☐ To understand the value of data mining in solving real-world problems.
- ☐ To gain understanding of algorithms commonly used in data mining tools.
- ☐ To develop ability for applying data mining tools to real-world problems

---

Unit-1:

Unit - I Introduction: Motivation: Important, Data type for Data Mining: Relational Databases Data Ware-Houses. Transactional Databases, Advanced Database System and Its Applications, Data Mining Functionalities Concept/Class Description, Association Analysis Classification & Prediction, Cluster Analysis, Outliner Analysis Classification of Data Mining Systems, Major Issues in Data Mining

Unit-2:

Data Warehouse and OLTP Technology for Data Mining: Differences between Operational Database Systems & Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Data Cube Technology, Emerging Scenario of Pattern Warehousing System

UNIT -3:

Data Pre-processing: Data Cleaning, Data Integration and Transformation, Data Reduction Discretization and Concept Hierarchy Generation. Data Mining Primitives Languages and System Architectures, Concept Description, Characterization and Comparison Analytical Characterization.

UNIT-4:

Mining Association Rules in Large Databases: Association Rule Mining Market Basket Analysis, Basic Concepts, Mining Single Dimensional Boolean Association Rules from Transactional Databases: The Apriori Algorithm, Generating Association Rules from Frequent Items, Improving the Efficiency of Apriori, other Algorithms & their Comparison, Mining Multilevel Association Rules, Multidimensional Association Rules, Constraint Based Association Rule Mining.

UNIT -5:

Classification & Predication and Cluster Analysis: Issues Regarding Classification & Predication, Different Classification Methods, Predication, Cluster Analysis, Major Clustering Methods, Currently Available Tools, Case Study.

RECOMMENDED BOOKS

• Data Mining: Concepts and Techniques, Han and Kamber, Morgan Kaufmann Publications.

• Data Mining Techniques, A. K. Pujari, Universities Press Pvt. Ltd .

COURSE OUTCOMES:

After completion of this course, the students would be able to:

CO 1. **Classify** various databases systems and data models of data warehouse.

CO2. **Compare** various methods for storing & retrieving data from different data sources/repository.

CO3**. Apply** pre-processing techniques for construction of data warehouse.

CO4. **Analyze** data mining for knowledge discovery & prediction.

CO5. **Explain** data mining methods for identification of association for transactional databases.

CO6. **Develop** various classification and clustering algorithms for data using data mining.

Department of Computer Science and Engineering

Theory of Computation

3150404

COURSE OBJECTIVES

- To understand computability, decidability, and complexity through problem solving.
- To analyse and design abstract model of computation & formal languages
- To understand and conduct mathematical proofs for computation and algorithms.

Unit-I

Introduction to Theory of Computation: Automata, Computability and Complexity, Alphabet, Symbol, String, and Formal Languages, Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and Mealy machines, Composite Machine, Conversion from Mealy to Moore and vice versa.

Unit-II

Types of Finite Automata: Non Deterministic Finite Automata (NDFA), Deterministic finite automata machines, conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Pumping lemma, applications, Closure properties of regular languages, 2 way DFA.

Unit-III

Grammars: Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, Rightmost and Leftmost derivations of Strings, ambiguity in grammar, simplification of context free grammar, killing null and unit productions, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar,. Chomsky Normal Form (CNF) and Greibach Normal Form (GNF).

Unit-IV

Turing Machine: Techniques for construction. Linear bounded automata, Church Turing Thesis, Encoding of Turing Machine, Universal Turing machine Multitape, multihead and multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem (PCB).

RECOMMENDED BOOKS

- Introduction to Automata Theory Language & Computation, Hopcroft & Ullman, Narosa Publication.
- Element of the Theory Computation, Lewis & Christors,Pearson.
- Theory of Computation, Chandrasekhar & Mishra,PHI.
- Theory of Computation, Wood, Harper &Row.
- Introduction to Computing Theory, Daniel I-A Cohen,Wiley.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Describe the basic concepts of switching and finite automata theory & languages.

CO2.Compute abstract models of computing and check their power to recognize the languages.

CO3. Analyse the grammar, its types, simplification and normal form.

CO4. Design mathematical models to prove properties of languages, grammars and automata.

CO5. Apply automata theory, languages and computation in engineering application.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	1			3	2								
CO2	1		3	3	2		3		2				1	2
CO3		1		2	2	3			2			2	2	2
CO4			1	1	1	2			2			2	1	2
CO5				1	2	2						1	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Theory of Computation

2150414(OLD)

COURSE OBJECTIVE

- To understand computability, decidability, and complexity through problem solving.
- To analyse and design abstract model of computation & formal languages
- To understand and conduct mathematical proofs for computation and algorithms.

Unit-I

Introduction to Theory of Computation: Automata, Computability and Complexity, Alphabet, Symbol, String, and Formal Languages, Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and Mealy machines, Composite Machine, Conversion from Mealy to Moore and vice versa.

Unit-II

Types of Finite Automata: Non Deterministic Finite Automata (NDFA), Deterministic finite automata machines, conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Pumping lemma, applications, Closure properties of regular languages, 2 way DFA.

Unit-III

Grammars: Types of grammar, context sensitive grammar, and context free grammar, regular grammar. Derivation trees, Rightmost and Leftmost derivations of Strings, ambiguity in grammar, simplification of context free grammar, killing null and unit productions, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar,. Chomsky Normal Form (CNF) and Greibach Normal Form (GNF).

Unit-IV

Push down Automata: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack, Example of PDA, deterministic and non-deterministic PDA, conversion of PDA into context free grammar and vice versa, CFG equivalent to PDA.

Unit-V

Turing Machine: Techniques for construction. Universal Turing machine Multitape, multihead and multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, decidability, decidable languages, undecidable languages, Halting problem of Turing machine & the post correspondence problem (PCB).

RECOMMENDED BOOKS

- Introduction to Automata Theory Language & Computation, Hopcroft& Ullman, Narosa Publication.
- Element of the Theory Computation, Lewis &Christors,Pearson.
- Theory of Computation, Chandrasekhar &Mishra,PHI.
- Theory of Computation, Wood, Harper &Row.
- Introduction to Computing Theory, Daniel I-A Cohen,Wiley.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. Explain the basic concepts of switching and finite automata theory & languages.

CO2. Relate practical problems to languages, automata, computability and complexity.

CO3. Construct abstract models of computing and check their power to recognize the languages.

CO4. Analyse the grammar, its types, simplification and normal form.

CO5. Interpret rigorously formal mathematical methods to prove properties of languages, grammars and automata.

CO6. Develop an overview of how automata theory, languages and computation are applicable in engineering application.

Department of Computer Science and Engineering

Artificial Intelligence & Neural Network

3150405

COURSE OBJECTIVES

- Identify the problems where AI is required and the different methods available.
- Compare and contrast different AI techniques available.
- Define and explain learning algorithms
- Learn knowledge representation techniques and their importance in AI.
- Familiarize students with neural networks and their architectures, focusing on learning laws and activation functions.

Unit: 1

Introduction: Need and Scope of Artificial Intelligence, History, Definition of Artificial Intelligence, Task and Objectives of Artificial Intelligence, Techniques of Artificial Intelligence. Artificial Intelligence Problems: Problems Definition, Problem Spaces and Production System. Characteristics of Production Systems, Types of Production System. Control Strategies, Example: water-jug, 8 – Puzzle, Cannibals & Missionaries problems.

Unit: 2

Agent: Introduction, Types of Agent, Searching techniques: Uninformed search, Breadth search and Depth first search, Depth limited Search, Informed (Heuristic) Search: Best-First Search, A* Algorithm, AO*, Simulated Annealing, Measure of performance and analysis of search algorithms, Constraint Satisfaction Problems, Adversarial Search.

Unit: 3

Knowledge Representation: Game playing, Definition and importance of Knowledge, Approaches to knowledge Representation, Issues in Knowledge Representation, Procedural and Declarative Knowledge, Knowledge Representation Techniques: Logics, Prepositional Logic, Predicate Logic, Semantic networks.

Unit: 4

Introduction to Neural Networks: Terminology and Comparison of Artificial Neural Networks vs. Biological Neural Networks. Learning Laws: Memory-Based Learning, Delta Learning, Perceptron Learning, Competitive Learning, and Boltzmann Learning Rule. Activation Functions: Binary Step, Linear, and Non-Linear Activation Functions.

Unit: 5

Models and Architectures of Neural Networks, Models of Neuron: McCulloch-Pitts Model, Hebb Net. Feedback Networks: Backpropagation Algorithm and Generalization, Single layer & Multi-Layer Perceptron.

RECOMMENDED BOOKS

- Russell, S.J. and Norvig, P., Artificial Intelligence: A Modern Approach, Pearson Education.
- Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", McGraw Hill.
- Simon Haykin, "Neural Networks and Learning Machines", 3rd edition, pearson education, 2008
- S. N. Sivanandam, S. N. Deepa, "Principles of Soft Computing", 2nd Edition

---

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Identify the AI based problems

CO2. Apply techniques to solve the AI problems

CO3. Define learning and explain various learning techniques

CO4. Discuss the concepts and architectures of Neural Networks.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2										3	2
CO2	3	3	3	2									3	2
CO3	3	2	2	1									2	3
CO4	3	3		2									3	3
CO5	2	2								2				2

1 - Slightly; 2 - Moderately; 3 – Substantially

Artificial Intelligence & Neural Network

3150405

List of Experiments

1. Write a program for Depth First Search
2. Write a program for Best First Search
3. Write a program to generate the output for A* algorithm
4. Write a program to solve water Jug problem using Heuristic functions
5. Write a program to show the Tic Tac Toe game from 0 and X
6. Write a program for expert system using Forward Chaining
7. Hands-on on Python for AI related problems like Neural Network, Genetic Algorithm, etc.
8. Project work as decided by Tutor. (all tools related to AI can be explored)

Department of Computer Science and Engineering

Programming Lab (Java Programming)

3150406

COURSE OBJECTIVES

- ☐ To understand fundamentals of Java programming such as variables, conditional and iterative execution, and methods.
- ☐ To understand fundamentals of object-oriented programming in Java, including defining Classes, invoking methods, using class libraries.
- ☐ To create a computer program to solve specified real world problems.

Unit 1

Introduction to Java Programming, Introduction to Java and its features, Java Development Kit (JDK) installation and setup, Java development environment (IDE) usage, Java syntax and basic programming concepts, Variables, data types, and operators, Control structures: decision-making and loops

Unit 2

Object-Oriented Programming in Java, Object-oriented programming (OOP) concepts: classes, objects, inheritance, polymorphism, and encapsulation, Java classes and objects, Constructors and methods, Inheritance and interfaces, Packages and access control.

Unit 3

Exception Handling and File Handling, Exception handling: try-catch blocks, multiple catch clauses, and exception hierarchy, Throwing and catching exceptions, File I/O operations: reading from and writing to files, Working with streams and readers/writers, File handling best practices and error handling

Unit 4

Java Collections Framework, Overview of Java Collections Framework (JCF), Lists, Sets, and Maps in JCF, ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, etc., Working with collections: adding, retrieving, updating, and deleting elements, Iterators and iterating over collections

Unit 5

Multithreading and Java GUI Programming, Multithreading concepts: threads, synchronization, and inter-thread communication, Creating and managing threads in Java, Thread synchronization and deadlock prevention, Introduction to Java GUI (Graphical User Interface) programming Event-driven programming and handling GUI events, Swing components and layout management

Reference Books

1. "Java: A Beginner's Guide" by Herbert Schildt (McGraw-Hill Education)
2. "Effective Java" by Joshua Bloch (Addison-Wesley Professional)
3. "Head First Java" by Kathy Sierra and Bert Bates (O'Reilly Media)
4. "Java: The Complete Reference" by Herbert Schildt (McGraw-Hill Education)
5. "Java Concurrency in Practice" by Brian Goetz et al. (Addison-Wesley Professional)

Course Outcomes

After completion of the course students would be able to:

CO1. Apply object-oriented programming principles, including inheritance, polymorphism, and encapsulation, to design and implement robust Java applications.

CO2. Utilize the Java Collections Framework to effectively manage and manipulate data structures, such as lists, sets, and maps.

CO3. Create interactive graphical user interfaces (GUI) using Java Swing components, incorporating event-driven programming to enhance user experience.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	22	2										3	2
CO2	2	2	3	2									3	2
CO3				2					2				2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Programming Lab (Java Programming)

3150406

List of Experiments

Experiment 1: Setting up the Java Development Environment

- Install the Java Development Kit (JDK) and an Integrated Development Environment (IDE).
- Write a simple "Hello, World!" program and execute it.

Experiment 2: Implementing Basic Control Structures

- Write a program that demonstrates the use of if-else statements for decision-making.
- Implement loops (for, while) to iterate over a set of numbers or perform a specific task.

Experiment 3: Creating and Manipulating Objects

- Design a class representing a student with relevant attributes and behaviors.
- Create multiple instances of the class and invoke methods to perform operations on the student objects.

Experiment 4: Inheritance and Polymorphism

- Create a base class and derived classes to showcase inheritance.
- Demonstrate polymorphism by invoking methods overridden in derived classes.

Experiment 5: Exception Handling

- Write a program that throws and catches different types of exceptions.
- Handle exceptions using try-catch blocks to prevent program termination.

Experiment 6: File Handling

- Read data from a text file and display its content.
- Write data to a file and verify the successful write operation.

Experiment 7: Working with Java Collections

- Create a collection (e.g., ArrayList) and perform operations like adding, retrieving, and removing elements.
- Iterate over a collection using iterators and demonstrate different collection classes.

Experiment 8: Multithreading

- Create multiple threads and execute them concurrently.
- Implement synchronization mechanisms to prevent thread interference.

Experiment 9: GUI Application Development

- Design a graphical user interface (GUI) using Swing components.
- Implement event handlers for GUI components, such as buttons or text fields.

Experiment 10: Comprehensive Project

- Design and implement a comprehensive Java project that incorporates concepts covered throughout the syllabus.
- Examples could include creating a student management system or a simple game using GUI elements.

Programming Lab (Java Programming)

3150406

List of Skill Based Mini Projects

1. Student Management System:
   - Design a console-based application to manage student information.
   - Implement functionalities like adding, deleting, and displaying student records.
2. Library Management System:
   - Create a program to manage library operations, including book borrowing, returning, and searching.
   - Implement data structures to store book records efficiently.
3. Calculator Application:
   - Develop a GUI-based calculator application using Swing components.
   - Implement basic arithmetic operations and handle user input.
4. File Encryption and Decryption:
   - Create a program to encrypt and decrypt files using encryption algorithms.
   - Provide options for the user to select the encryption method and specify the file to encrypt/decrypt.
5. Quiz Application:
   - Develop a quiz application that presents multiple-choice questions to the user.
   - Implement a scoring system and display the result at the end of the quiz.
6. Bank Account Management System:
   - Design a program to manage bank accounts, including features like account creation, deposit, withdrawal, and balance inquiry.
   - Implement object-oriented concepts to model bank accounts and transactions.
7. Contact Management Application:
   - Develop a console-based application to manage contacts.
   - Implement functionalities like adding contacts, searching by name, and displaying contact details.
8. Tic-Tac-Toe Game:
   - Create a GUI-based Tic-Tac-Toe game using Swing components.
   - Implement game logic to handle player turns and determine the winner.
9. Weather Forecast Application:

- Develop a program that retrieves weather data from an API and displays it to the user.
- Implement features like displaying current weather, forecast for multiple days, and location-based search.

10. Online Shopping System:
    - Design a simple online shopping system with features like browsing products, adding items to the cart, and placing orders.
    - Implement shopping cart functionality and user authentication.