Course Code	Course Title						Core/Elective
U24IT501	AUTOMATA THEORY, LANGUAGES AND COMPUTATION						Core
Prerequisite	Hours Per Week				CIE	SEE	Credits
Discrete Mathematics	L	T	D	P			
	3	1	-	-	40	60	4

Course Objectives

Develop ability to

1. Develop a formal notation for strings, languages and machines.
2. Understand Regular Expression and algebraic laws.
3. Design context free grammars and PDA, to generate strings from a context free language and convert them into normal forms.
4. Identify the hierarchy of formal languages, grammars and machines.
5. Distinguish between computability and non-computability and Decidability and undecidability.

Course Outcomes

At the end of the course, student would be able to

1. Gain knowledge of the various abstract machines.
2. Use regular languages and regular expression for constructing different finite state machines.
3. Understand and design different types of grammars.
4. Construct Push down Automata.
5. Construct Turing Machine.

UNIT-I

Introduction: Introduction to Finite Automata, Structural Representations, Basic Concepts of Automata Theory – Symbol, Alphabets, Strings, Languages, Kleene Closure, Positive Closure.
Finite Automata without Output: Deterministic Finite Automata, Nondeterministic Finite Automata, Finite Automata with Epsilon-Transitions, Conversion of NFA to DFA.

UNIT-II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of regular sets, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages: Pigeon Hole Principle, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Minimization of Automata- Equivalence, Myhill-Nerode Theorem

UNIT-III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Ambiguity in Grammars.
Push down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state & empty stack, Deterministic PDA Conversion of CFG to PDA, Conversion of PDA to CFG

UNIT-IV

Normal Forms for Context- Free Grammars: Grammar Simplification, Chomsky Normal form, Greibach Normal form.

Context-Free Languages: Statement of pumping lemma, Applications pumping lemma

Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's

UNIT-V

Turing Machines: Introduction, Techniques for construction of Turing machines, Modifications of TM, TM as enumerator, Restricted TM.

Undecidability: Recursive and Recursively enumerable languages, Halting problem, Rice Theorem, Post's correspondence problem, P,NP,NP-Complete, NP-Hard Problems, Chomsky's Hierarchy–Regular grammars, Unrestricted grammar, CSL

Suggested Reading:

1. Introduction to Automata Theory, languages, and Computation, John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Pearson Education India, 5th Edition,2017.
2. Automata and Computability, Undergraduate Texts in Computer Science,Dexter C. Kozen, Springer, 2015.
3. Introduction to the Theory of Computation, Michael Sipser, PWS Publishing 3rd Edition,2014.
4. Introduction to Languages and the Theory of Computation, John Martin, Tata McGraw Hill, 3rd Edition, 2002.

Course Code	Course Title						Core/Elective
U24IT502	EMBEDDED SYSTEMS & INTERNET OF THINGS						CORE
Prerequisite	Hours Per Week				CIE	SEE	Credits
COM	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1.Learn the internal architecture and programming of an embedded processor.
2. Introduce interfacing I/O devices to the processor.
3. Introduce the evolution of the Internet of Things (IoT).
4. Build a small low-cost embedded and IoT system using Arduino/Raspberry Pi/open platform.
5.Apply the concept of Internet of Things in real world scenario

Course Outcomes

At the end of the course, student would be able to

1. Learn the internal architecture and programming of an embedded processor.
2. Interface I/O devices to the processor.
3. Present the Internet of Things' (IoT) progress.
4. Use the Arduino, Raspberry Pi, and open platform, to construct a small, inexpensive embedded and Internet of Things system.
5. Use the Internet of Things concept in a practical setting.

UNIT – I

8-Bit Embedded Processor: 8-Bit Microcontroller, Architecture, Instruction Set and Programming, Programming Parallel Ports, Timers and Serial Port, Interrupt Handling.

UNIT – II

Embedded C Programming: Memory And I/O Devices Interfacing, Programming Embedded Systems in C, Need For RTOS, Multiple Tasks and Processes, Context Switching, Priority Based Scheduling Policies.

UNIT- III

IoT and Arduino Programming: Introduction to the Concept of IoT Devices, IoT Devices Versus Computers, IoT Configurations, Basic Components, Introduction to Arduino, Types of Arduino, Arduino Toolchain, Arduino Programming Structure, Sketches, Pins Input/Output from Pins Using Sketches, Introduction to Arduino Shields, Integration of Sensors and Actuators with Arduino.

UNIT – IV

IoT Communication and Open Platforms: IoT Communication Models and APIs, IoT Communication Protocols, Bluetooth, WiFi, ZigBee, GPS, GSM modules, Open Platform (like Raspberry Pi), Architecture, Programming, Interfacing, Accessing GPIO Pins, Sending and Receiving Signals Using GPIO Pins, Connecting to the Cloud.

UNIT - V

Applications Development: Complete Design of Embedded Systems, Development of IoT Applications, Home Automation, Smart Agriculture, Smart Cities, Smart Healthcare.

Suggested Readings:

1. Raj Kamal, Embedded Systems – Architecture, Programming and Design, 2nd Edition, TMH, 2008.
2. Andrew N. Sloss, Dominic Symes, Chris Wright, ARM Systems Developer's Guides – Designing & Optimizing System Software, Elsevier, 2008.
3. Vijay Madisetti and Arshdeep Bahga, Internet of Things: A Hands-On Approach, VPT edition1, 2014.

Course Code	Course Title						Core/Elective
U24CD501	ARTIFICIAL INTELLIGENCE						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DM, M-III	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand the importance of the field of AI by discussing its history and various applications.
2. Learn about one of the basic applications of A.I, search state formulations
3. Learn knowledge representation implementation.
4. Learn how to reason when an agent has only uncertain information about its task.
5. Know various supervised and unsupervised learning algorithms

Course Outcomes

At the end of the course, student would be able to

1. Illustrate basic principles of AI in solutions that require problem solving, search, inference
2. Demonstrate understanding of steps involved in building of intelligent agents, expert systems, Bayesian networks
3. Differentiate between learning paradigms to be applied for an application
4. Demonstrate Expert system its utilization
5. Illustrate AI application machine learning & its types.

Unit-I:

Introduction to Artificial Intelligence: Foundations of artificial intelligence (AI), Intelligent Systems, History of AI, Subareas of AI, Applications, Structure of Agents, Types of agents, AI problems, Agents and Environments.

UNIT -II:

Basic Search Strategies: Problem Spaces, Uninformed Search (Breadth-First, Depth First Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A*), Constraint Satisfaction (Backtracking, Local Search)

Advanced Search: Constructing Search Trees, Stochastic Search, A* Search Implementation, Minimax Search, Alpha-Beta Pruning.

UNIT-III:

Reasoning: Knowledge based agent, Propositional Logic, Inference, Predicate logic (first order logic),Inference, Resolution, Frames, Semantic Nets

Uncertainty :Basic probability, Bayes rule, Naive Bayes, Belief networks, Inference in Bayesian Network.

UNIT-IV

Expert System and Applications: Introduction, Phases in Building Expert Systems, Expert System Architecture, Applications.

UNIT-V

Machine-Learning Paradigms: Introduction, Machine Learning Systems, Supervised and Unsupervised Learning, Reinforcement learning with applications, Natural Language processing, Speech recognition.

Suggested Readings:

1. Artificial Intelligence – A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education Press. Fourth Edition.
2.Artificial Intelligence, Saroj Kaushik, Cengage Learning, 2011.
3.Artificial Intelligence, Kevin Knight, Elaine Rich, B.Nair, McGraw Hill, 3rd Edition, 2008.

Course Code	Course Title						Core/Elective
U24IT503	**DATA VISUALIZATION**						**Professional Elective-I**
Prerequisite	**Contact Hours Per Week**				**CIE**	**SEE**	**Credits**
Python Programming	**L**	**T**	**D**	**P**			
	3	**-**	**-**	**-**	**40**	**60**	**3**

Course Objectives

Develop ability to

1. Apply the principles and concepts of data visualization to represent information effectively.
2. Classify data types, data structures, and human perception factors that influence visualization design..
3. Select appropriate visualization techniques for different data types and analytical requirements
4. Develop data visualizations using Matplotlib by creating and customizing a variety of plots.
5. Create statistical and geospatial visualizations using Seaborn and Folium for data analysis and interpretation.

Course Outcomes

At the end of the course, student would be able to

1. Describe the concepts, objectives, process, and applications of data visualization.
2. Classify data types, data structures, preprocessing methods, and perception principles used in visualization
3. Analyze visualization foundations and **select** appropriate visualization techniques for different datasets.
4. Apply Matplotlib to generate and customize different types of data visualizations
5. Create statistical and geospatial visualizations using Seaborn and Folium for data interpretation.

UNIT I

Introduction to Data Visualization: Definition, objectives, importance and applications, Concepts of Visualization. History of Visualization, Relationship between Visualization and Other Fields, The Visualization Process, The Role of Cognition, Scatterplots.

UNIT II

Data Foundations and Human Perception:

Data Foundations: Types of Data; Data Structures, Data Preprocessing, Data Sets.

Human Perception: Perception, Information Processing, Cognition Perception in Visualization.

UNIT III

Visualization Foundations and Techniques

Visualization Foundations: Visualization Process; Graphical Symbols; Visual Variables; Visualization Taxonomies.

Visualization Techniques: One-Dimensional, Two-Dimensional, Three-Dimensional, Dynamic and Geospatial Data Visualization; Visualization of Point, Line, and Area Data; Selection of Appropriate Visualization Techniques.

UNIT IV

Introduction to Matplotlib: Installation and configuration; Anatomy of a plot; Basic plots using Matplotlib (Line Plot, Bar Chart, Scatter Plot, Histogram, Pie Chart); Plot customization (Titles, Labels, Legends, Grid, Styles); Specialized visualizations (Box Plot, Area Chart, Stem Plot, Error Bar Plot); Advanced visualizations using Waffle Charts and Word Clouds**.**

UNIT V

Introduction to Seaborn: Seaborn functionalities and usage, Spatial Visualizations and Analysis in Python with Folium, Case Study.

Suggested Readings:

1. *Interactive Data Visualization: Foundations, Techniques, and Applications*, Matthew O. Ward, Georges Grinstein, Daniel Keim, 2nd Edition, CRC Press, Taylor & Francis Group.
2. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*, Wes McKinney, 3rd Edition, O'Reilly Media, 2022.
3. *Python Data Science Handbook: Essential Tools for Working with Data*, Jake VanderPlas, 1st Edition, O'Reilly Media, 2016.
4. *Hands-On Data Visualization with Pandas and Matplotlib*, Thomas A. Caswell, Brian E. Granger, and Matplotlib Development Team, O'Reilly Media.
5. *Data Visualization with Python*, IBM Skills Network (Coursera Learning Materials).

Course Code	Course Title						Core/Elective
U24IT5L1	EMBEDDED SYSTEMS & INTERNET OF THINGS LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DLD	L	T	D	P			
	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1.Familiarize with interfacing LEDs and switches using 8051 Microcontroller.
2.Aquaint with interface controls using 8051.
3.Explore design and development of an embedded system.
4.Know the interfacing programs using Python.
5.Understand the applications using Raspberry Pi and Arduino.

Course Outcomes

At the end of the course, student would be able to

1. Possess the passion for acquiring programming skills in using different tools.
2.Able to design and develop embedded systems (hardware, peripherals and firmware).
3.Write code for different forms of interfacing devices.
4.Develop python programs that run on Raspberry Pi4.
5.Interface Sensors and Actuators with Raspberry Pi4.

List of Experiments

A. Interfacing Programs using embedded C on ARM Microcontroller Kit:

1. Program to perform arithmetic operation (addition, subtract, multiply, division).
2. Program to interface8-Bit LED and switch interface.
3. Program to implement Buzzer interface on IDE environment.
4. Stepper motor interface.
5. ADC &Temperature sensor LM35interface.
6. Program to produce show the Time delay.

B. Internet of Things (IoT) Experiments to write and test on Raspberry Pi.

1.Switching LED on/off from Raspberry Pi Console.
2.Interfacing an LED and Switch with Raspberry Pi.
3.Interfacing a Light Sensor with Raspberry Pi.
4.Interfacing Smoke Sensor to give alert message to fire department.
5.Implementation of Traffic Light System based on density, to decrease congestion.
6.Implementation of Home Automation System using WiFi Module.

Suggested Readings:

1.Kenneth J.Ayala, "The 8051 Microcontroller", 3rd Edition, Thomson, 2014.
2.ArshdeepBahga, Vijay Madisetti, "Internet of Things: A Hands-on Approach", Universities Press, 2014.

Course Code	Course Title						Core/Elective
U24CD5L1	ARTIFICIAL INTELLIGENCE LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Python Programming	L	T	D	P			
	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1. Apply programming skills to formulate the solutions for computational problems.
2. Implement first order predicate calculus using Prolog.
3. Understand python library scikit-learn for building machine learning models.
4. Select and apply relevant AI tools for the given problem.
5. Understand the searching techniques and its implementation.

Course Outcomes

At the end of the course, student would be able to

1. Develop solutions for informed and uninformed search problems in AI.
2. Apply reasoning in first order logic using Prolog.
3. Apply python libraries to synthesize information and develop supervised learning models.
4. Develop a case study in multidisciplinary areas to demonstrate the use of AI.
5. Analyse and develop real time applications by using AI tools.

List of Experiments

1. Python program to implement List operations.
2. Implementation of Nested List, Length, Concatenation, Membership, Iteration, Indexing and Slicing functions.
3. Implementation of List methods: Add, Append, Extend & Delete.
4. Program to eliminate punctuations in a given string.
5. Sorting a sentence in an alphabetical order.
6. Implementation of Breadth First Search Traversal Technique.
7. Depth First Search Traversal Technique Implementation.
8. Implementation of Water Jug Problem using Heuristic Method.
9. Implementation of Travelling Salesman Problem using Python.
10. Writing simple facts for the statements and querying using Prolog.
11. Demonstration of Family tree using Prolog using conditional statements.
12. CASE STUDY
    a. Simple Chatroom using Python
    b. Hospital Management System Project in Python

Suggested Readings:

1. Artificial Intelligence – A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education Press,
Third edition.
2. Artificial Intelligence, Saroj Kaushik, Cengage Learning, 2011.

Course Code	Course Title						Core/Elective
U24IT5L3	DATA VISUALIZATION LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DBMS	L	T	D	P			
	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1. Effective use of Business Intelligence (BI) technology (Tableau/Power BI/Google Data Cloud Studio) to apply data visualization.
2. Discern patterns and relationships in the data.
3. Build Dashboard applications.
4. Communicate the results clearly and concisely.
5. Work with different formats of data sets

Course Outcomes

At the end of the course, student would be able to

1. Learn how to import data into Tableau.
2. Enumerate Tableau concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties.
4. Create a Dashboard that links multiple visualizations.
5. Use graphical user interfaces to create Frames for providing solutions to real-world problems

List of Experiments

1. Understanding Data, What is data, where to find data and Creating First Visualization in Python.
2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts (line, bar charts, Tree maps), Using the Show me panel.
3. Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.
4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, and Formatting specific parts of the view.
5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.
6. Structuring your data, Sorting and filtering Tableau data, and Pivoting Tableau data.
7. Creating Dashboards & Storytelling, creating your first dashboard and Story, and Design for different displays, and publish your visualization.
8. Creating custom charts, cyclical data, circular area charts, and Dual Axis charts.
9. Advanced Visualization Tool Power BI: Using Filters, Using the Detail panel, using the Size panels, customizing filters, And customizing tooltips and formatting your data with colors.
10. Visualization data supported chart types in Power BI, Map Visualizations, Color palettes in Charts, Loading Shapes, text boxes, and images.

Suggested Reading:

1. Data Visualization in Python by Daniel Nelson, 2020 Stack Abuse.
2. Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software by Daniel G Murray, WILEY
3. Microsoft Power BI cookbook, Brett Powell, 2nd edition.
4. Learning Tableau by Joshua Milligan,2019 PACKT publishing.

Course Code	Course Title						Core/Elective
U24IT5P1	INTERNSHIP						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming Languages	L	T	D	P			
	-	-	-	2	50	-	1

Course Objectives

Develop ability to

1. Produce an accurate record of work performed during the Internship
2. Apply engineering knowledge to a problem in industry
3. Produce a technical report
4. Discuss work in a team environment, if relevant to the project
5. Conduct herself/himself responsibly, safely, and ethically in a professional environment

Course Outcomes

At the end of the course, student would be able to

1. Design and develop a small and simple product in hardware or software.
2. Complete the task or realize a pre specified target, with a specified scope.
3. Learn to find alternate viable solutions for a given problem and evaluate these alternatives with reference to pre specified criteria.
4. Gain knowledge of working practices within industrial / R&D environments.
5. Implement the selected solution and document the same.

Guidelines:

Internship is introduced as part of the curriculum for encouraging students to work on problems of interest to industries. A batch of three students will be attached to a person from the Government or Private Organizations /Computer Industry/Software Companies/R&D Organization for a period of 4 weeks. This will be during the summer vacation following the completion of the IV Semester. One faculty coordinator will also be attached to the group of 3 students to monitor the progress and to interact with the industry coordinate (person from industry). The course schedule will depend on the specific internship/training experience. The typical time per topic will vary depending on the internship

1. Overview of company/project
2. Safety training
3. Discussions with project teams
4. Background research, review of documents, white papers, and scientific papers
5. Planning, designing, and reviewing the planned work
6. Executing the plans
7. Documenting progress, experiments, and other technical documentation
8. Further team discussions to discuss results
9. Final report writing and presentation

After the completion of the internship, each student will be required to:

1. Submit a brief technical report on the internship undergone.
2. Present the work through a seminar talk (to be organized by the Department)

Internship will be evaluated for 50 marks as CIE based on Punctuality in Attendance (10), Maintenance of Diary on daily work done (20), seminar presentation / viva voce followed by a report submission (20) to a committee consisting of Head of the Department, Senior Faculty and Faculty Advisor / Supervisor for evaluation.

Note: Students have to undergo summer internship of 4 weeks at the end of semester IV and credits will be awarded after evaluation in V semester.

Course Code	Course Title						Core/Elective
U24IT601	FULL STACK DEVELOPMENT						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Java Programming	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Work with Webserver and Java Script
2. Gain knowledge of HTTP services and the DNS Modules.
3. Work with MongoDB and the associated connection objects.
4. Learn the Express and the Angular JS.
5. Understand the ReactJS and its Lifecycle.

Course Outcomes

At the end of the course, student would be able to

1. Understand Full stack components for developing web application.
2. Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
3. Use MongoDB data base for storing and processing huge data and connects with NodeJS application.
4. Design faster and effective single page applications using Express and Angular.
5. Create interactive user interfaces with react components.

UNIT-I

Introduction to Full Stack Development:

Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks.

UNIT-II

Node.js:

Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the OS Module, Using the Util Module, Using the DNS Module, Using the crypto Module.

UNIT-III

MongoDB:

Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV

Express and Angular:

Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT-V

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

Suggested Reading:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

Course Code	Course Title						Core / Elective
U24CM603	COMPUTER NETWORKS						Professional Elective & Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
C	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand communication in modern network architectures from a design and performance perspective.
2. Understand Data Transmission standards and MAC protocols.
3. Introduce the protocols functionalities in Network Layer and Transport Layer.
4. Understand DNS and supportive application protocols.
5. Provide basic concepts of Cryptography.

Course Outcomes:

1. Explain data communication fundamentals, OSI and TCP/IP models, bandwidth utilization techniques, and transmission media.
2. Apply data link layer protocols for error control, flow control, framing, and multiple access.
3. Analyze network layer addressing, switching techniques, routing, and Internet control protocols.
4. Evaluate transport layer protocols and congestion control mechanisms for reliable end-to-end communication.
5. Develop basic network applications using socket programming and explain core application layer services.

UNIT-I

Overview: Data communication, Networks, The Internet, Protocols and Standards, Layered tasks OSI Model and layers in OSI model, TCP/IP protocol suite, Addressing, Transmission Modes.

Techniques for Bandwidth utilization: Multiplexing - Frequency division, Wavelength division, Synchronous and Statistical Time division multiplexing,

UNIT-II

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction**,** Introduction to Error Detection and error correction, Block coding, CRC.

Data link control: Framing, Flow Control and Error control, Protocols-Simplest Protocol, Stop-and-Wait Protocol, Stop-and-Wait ARQ, Go-Hack-N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking.

Multiple Access: Random Access-Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA,

UNIT-III

Network Layer-Switching Techniques: Circuit switching networks, Packet Switching networks Virtual circuit networks. Logical addressing, Inter Networking, IPV4, IPV6, Tunneling.

Internet Control Protocols: Address Mapping -ARP, RARP, BOOTP and DHCP.
ICMP, IGMP. Delivery, Forwarding, Unicast Routing protocol,

UNIT-IV

Transport Layer: Process to Process Delivery, UDP, TCP, Data Traffic, Congestion, Congestion Control, Congestion Control in TCP. Quality of Service, Techniques to improve QoS.

UNIT-V

Socket Programming: Socket address, Elementary socket system calls, Advanced socket system calls.

Application Layer: Domain Name Space (DNS), EMAIL, SNMP, and Multimedia

Text Books: 1. Data Communication and Networking, 4th Edition, Behrouz A.Forouzan, McGrawHill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. W. Richard Stevens, Unix Network Programming, Prentice Hall / Pearson Education, 2009.

Course Code	Course Title						Core/Elective
U24CM608	MACHINE LEARNING						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Python Programming	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand the basic concepts of machine learning and range of problems that can be handled by machine learning
2. Analyze the concepts learning and decision tree induction
3. Understand the concepts of unsupervised learning and evolutionary learning
4. Understand the concepts of ensemble learning, dimensionality reduction and clustering
5. Gain experience of doing independent study and research

Course Outcomes

At the end of the course, student would be able to

1. Design and implement machine learning solutions imply variant techniques,
2. Calculate Statistical measurements of the given data.
3. Analyze and identify the best algorithm matches for a given dataset.
4. Evaluate and interpret the results of the machine learning algorithms
5. Design and implement various machine learning algorithms in a range of real-world applications

Unit - I

Introduction: Learning, Types of Machine Learning.
Concept learning: Introduction, Version Spaces and the Candidate Elimination Algorithm.
Some Basic Statistics: Averages, Variance and Covariance, The Gaussian, The Bias-Variance, Bayes theorem. Bayes Optimal Classifier, Naïve Bayes Classifier.

Unit - II

Learning with Trees: Constructing Decision Trees, CART, Classification Example
Linear Discriminants: The Perceptron, Linear Separability, Linear Regression
Multilayer Perceptron (MLP): Going Forwards, Backwards, MLP in practices, Deriving back.

Unit - III

Clustering: Introduction, Similarity and Distance Measures, Outliers, Hierarchical Methods, Partitional Algorithms, Clustering Large Databases, Clustering with Categorical Attributes, Comparison
Graphical Models: Bayesian networks, Approximate Inference, Making Bayesian Networks, HiddenMarkov Models, The Forward Algorithm.

Unit - IV

Evolutionary Learning: Genetic Algorithms, Genetic Operators, Genetic Programming.
Ensemble learning: Boosting, Bagging.
Dimensionality Reduction: Linear Discriminant Analysis, Principal Component Analysis

Unit – V

Reinforcement Learning: Introduction, the learning task, Q–learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.

Suggested Readings:

1. Machine Learning - An Algorithmic Perspective, Stephen Marsland, CRC Press, 2009
2. Machine Learning, Tom M. Mitchell, Mc Graw Hill, 1997
3. Data Mining, Margaret H Dunham, Pearson Edition., 2003.
4. Data Mining for Business Intelligence, GalitShmueli, Nitin R Patel, Peter C Bruce, Wiley India Edition, 2007
5. Pattern Recognition, RajjanShinghal, Oxford University Press, 2006.

Course Code	Course Title				Core/Elective		
U24IT6L1	FULL STACK DEVELOPMENT LAB				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Java Programming	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1. Build software development skills using java programming for real world applications.
2. Implement frontend and backend of an application
3. Create Java application programs using sound OOP practices such as interfaces, exception handling multithreading.
4. Understand fundamentals of object-oriented programming in Java.
5. Implement classical problems using java programming.

Course Outcomes

At the end of the course, student would be able to

1. Design flexible and responsive Web applications using Node JS, React, Express and Angular.
2. Perform CRUD operations with MongoDB on huge amount of data.
3. Develop real time applications using react components.
4. Use various full stack modules to handle http requests and responses.
5. Implement the components of ReactJS.

List of Experiments

1. Create an application to setup Node JS environment and display “Hello World”.
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response using NodeJS.
5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Write a program to create a voting application using React JS.

Suggested Readings:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley,2019.
2.Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

Course Code	Course Title						Core / Elective
U24CM6L3	COMPUTER NETWORKS LAB						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
C	-	-	-	3	25	50	1.5

Course Objectives:

1. To familiarize students with the basic principles, protocols, and layered architecture of computer networks.
2. To reinforce the theory learned in the classroom via hands-on practice.
3. Develop practical skills in network configuration and management
4. To design and implement client-server applications using protocols such as TCP and UDP.
5. To develop simple socket programming applications in languages like C, Java, or Python.

Course Outcomes:

1. Explain and apply basic networking concepts, layered architecture, and protocol operations in practical scenarios.
2. Configure and troubleshoot network devices and design small-scale network topologies.
3. Develop and implement client–server communication using socket programming in TCP and UDP.
4. Simulate and analyze network protocols and topologies using simulation tools .
5. Capture, inspect, and analyze network traffic using tools such as Wireshark to understand protocol behavior and performance metrics.

Sr No	Experiment No.	NAME OF THE EXPERIMENT
01	Exp-01	**Network Devices and Networking Commands basics**
02	Exp-02	**Using Cisco Packet Tracer (CPT)** **Build: The Network,** Design the topologies, Connect 4 computers in the LAN using Switch, Configure IP addressing on all PC using 192.168.1.0/24 network, Check connectivity between all the PÇ's using Ping command, traceroute, Configure IP Address as diagram & rules, Verify the interface status
03	Exp-03	**Configuring Static Routing using Packet Tracer** a) Assign IP address according to diagram b) Make sure that interfaces used should be in UP UP state c) Configure static routing d) Verify routing table and reachability between LAN's (Using PING & TRACE commands)
04	Exp-04	**Configuring Default Routing using Packet Tracer** a) Design the topology b) Assign IP address according to diagram c) Make sure that interfaces used should be in UP-UP state d) Configure Default route used on R1 & R3, Static routing on R2 e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)

05	**Exp-05**	**Configuring Dynamic Routing using Packet Tracer** a) Design the topology b) Assign IP address according to diagram c) Make sure that interfaces used should be in UP UP state d) Configure Dynamic Routing using EIGRP 100 e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)
06	**Exp-06**	a. Write a Program to demonstrate connectionless (UDP) b) Echo Service Using Well Known / Standard Port. **c)** Write a Program to demonstrate connectionless (UDP) Date and Time Service Using Well Known / Standard Port.
07	**Exp-07**	a. Write a Program to Demonstrate Connection oriented (TCP) Echo Service Using Well Known / Standard Port b. Write a Program to Demonstrate Connection oriented (TCP) Date and Time Service Using Well Known / Standard Port
08	**Exp-08**	1. Write a program to implement echo service using RPC. 2. Write a program to implement date and time service using RPC. Program name:day.x 3. Write a program to implement addition of two numbers using RPC.
09	**Exp-09**	Write a Program to Implement Mini DNS Service
10	**Exp-10**	Demonstrate by witing a program on Bit Stuffing/ Character stuffing, Link state routing and Distance Vector routing algorithms
11	**Exp-Extra**	Beyond the syllabus: NS2 simulations for Network analysis

Course Code	Course Title						Core/Elective
U24CM6L4	MACHINE LEARNING LAB						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Python Programming	L	T	D	P			
	-	-	-	3	25	50	1.5

Course Objectives

Develop ability to

1. Make use of Data sets in implementing the machine learning algorithms
2. Implement the machine learning concepts and algorithms in any suitable language of choice.

Course Outcomes

At the end of the course, student would be able to

1. Understand the implementation procedures for the machine learning algorithms.
2. Design Java/Python programs for various Learning algorithms.
3. Apply appropriate data sets to the Machine Learning algorithms.
4. Identify and apply Machine Learning algorithms to solve real world problems.

List of Experiments

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

4. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.

8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

Course Code	Course Title						Core/Elective
U24IT6L2	COMPETITIVE CODING						Core
Prerequisite	Hours Per Week				CIE	SEE	Credits
DS, DAA, Java & Python	L	T	D	P			
	3	-	-	-	50	-	1

Course Objectives

Develop ability to

1. Develop problem-solving skills using structured programming concepts and logical thinking.
2. Understand and implement fundamental data structures such as arrays, stacks, queues, linked lists, trees, and graphs.
3. Apply algorithmic techniques including recursion, searching, sorting, dynamic programming, and greedy methods for solving computational problems.
4. Design and implement object-oriented solutions using classes, objects, inheritance, polymorphism, and interfaces.
5. Build practical programming ability through real-world applications such as games, shopping carts, and notification systems using Python and Java.

Course Outcomes

At the end of the course, student would be able to

1. Develop programs using basic programming constructs such as variables, loops, conditional statements, functions, and random number generation.
2. Implement linear and non-linear data structures including arrays, linked lists, stacks, queues, trees, and graphs to solve computational problems.
3. Apply algorithm design techniques such as recursion, backtracking, searching, sorting, and dynamic programming for efficient problem solving.
4. Design object-oriented applications using classes, objects, encapsulation, inheritance, interfaces, and polymorphism.
5. Build and test real-world software applications with modular design, reusable code, and user interaction using Python and Java.

1. Implement a program where the computer generates a random number between 1 and 100. The user has to guess the number within a limited number of attempts.(Python)

- If guess is too high → print "Too High"
- If guess is too low → print "Too Low"
- If correct → print "Correct Guess" and exit
- If attempts are exhausted, display the correct number..

Input Format:
User enters guesses
Output Format:
Messages based on guess
Key Ideas:

i. Use random.randint()
ii. Use loop with attempt counter
iii. Conditional checks
iv. Break when correct

2. Implement a program to move all zeroes in an array to the end while maintaining the order of non-zero elements. (Python)

Input Format:
First line: n
Second line: array elements
Output Format:
Modified array with zeroes at end
Key Ideas: Two Pointers, In-place Swapping

3. Write a program to reverse a singly linked list. (Java)

Input Format:

Number of nodes
Node values
Output Format:
Reversed linked list
Key Ideas: Pointer Manipulation, Iteration

4. **Implement a queue using two stacks. (Java)**
   Input Format:
   Number of operations
   Enqueue / Dequeue commands
   Output Format:
   Queue after operations
   Key Ideas: Stack Transfer Logic

5. **Generate all permutations of a string recursively. (Python)**
   Input Format:
   n
   Set elements
   Output Format:
   All subsets
   Key Ideas: Recursion, Backtracking

6. **Find the K most frequent elements (Python)**
   Input Format:
   n
   Array elements
   k
   Output Format:
   Top k frequent values
   Key Ideas: Heap, Frequency Map

7. **Validate a Binary Search Tree by taking an example (Java)**
   Input Format:
   Tree nodes
   Output Format:
   True / False
   Key Ideas: Inorder Traversal, Range Check

8. **Find the number of ways to climb n stairs. (Python)**
   Input Format:
   Integer n
   Output Format:
   Number of ways
   Key Ideas: Fibonacci DP

9. **Online Shopping Cart (Java/Python)**
   Problem: (Multiple Objects + OOP Design)
   Create classes:
   1. Product →name, price
   2. Cart → List of products

   Add functionality to:
   Add product
   Remove product
   Calculate total bill

Key Ideas
Object composition
Use List inside class
Methods to manage data

10. Notification System (Java)

Problem
Create classes:
1. Email Notification.
2. SMSNotification
3. Push Notification

Each should implement a method send(message)
Write a program to call the same method for different notification types.
Output format:
Display notification message
Key Ideas
Same interface(send)
Different behaviour
Real-world abstraction

11. Write a program to detect whether a given graph contains a cycle. (Java/Python)

Input Format
First line contains two integers V and E representing the number of vertices and edges.
The next E lines contain two integers u and v, representing an edge between vertex u and vertex v.
Output Format
Print Cycle Detected if the graph contains a cycle.
Otherwise, print No Cycle.
Key Idea
Use **Depth First Search (DFS)** or **Breadth First Search (BFS)** to traverse the graph and detect back edges.
For an undirected graph:
If a visited node is reached again through a different path (not parent), a cycle exists.

12. Hospital Management System (Java)

Create classes:
1. Patient
2. Doctor
3. Appointment

Implement booking and display details.
Input Format:
Patient and doctor details
Output Format:
Appointment confirmation
Key Ideas:
Multiple classes
Object interaction
Real-world design

13. Find shortest path from a source node in a weighted graph.(Python)
Key Ideas: Graph, Priority Queue, Greedy

14. Library System (Real-World OOP)

Problem
Create classes:

1. Book→title, author
2. Library→ collection of books.

Add features: Add book

1. Issue book.
2. Return book.

Key Ideas

Interaction between classes
Maintain State
Use lists/dictionaries

Course Code	Course Title						Core/Elective
U24IT6P1	MINI PROJECT						CORE
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming Languages	L	T	D	P			
	-	-	-	4	50	50	2

Course Objectives

Develop ability to

1. Enhance practical and professional skills.
2. Familiarize tools and techniques of systematic literature survey and documentation
3. Expose the students to industry practices and team work.
4. Encourage students to work with innovative and entrepreneurial ideas.
5. Make students evaluate different solutions based on economic and technical feasibility

Course Outcomes

At the end of the course, student would be able to

1. Formulate a specific problem and give solution
2. Develop model/models either theoretical/practical/numerical form
3. Solve, interpret/correlate the results and discussions
4. Conclude the results obtained
5. Write the documentation in standard format

Guidelines:

1. As part of the curriculum in the VI- semester of the programme each student shall do a mini project, generally comprising about three to four weeks of prior reading, twelve weeks of active research, and finally a presentation of their work for assessment.
2. Four students will be allotted to one faculty supervisor for mentoring.
3. Mini projects should present students with an accessible challenge on which to demonstrate competence in research techniques, plus the opportunity to contribute something more original.
4. Mini projects shall have inter-disciplinary/ industry relevance.
5. The students can select a mathematical modelling based/Experimental investigations or Numerical modelling
6. All the investigations should be clearly stated and documented with the reasons/explanations.
7. The mini-project shall contain a clear statement of the research objectives, background of work, literature review, techniques used, prospective deliverables, and detailed discussion on results, conclusions and reference.
8. The CIE shall include reviews and the preparation of report consisting of a detailed problem statement and a literature review.
9. The preliminary results (if available) of the problem may also be discussed in the report.
10. The work has to be presented in front of the PRC committee which consists of one Supervisor and a minimum of two faculty members from the respective Department of the Institute.

U24IT504	COMPUTER GRAPHICS						CORE
Prerequisite	**Contact Hours Per Week**				**CIE**	**SEE**	**Credits**
C Language	**L**	**T**	**D**	**P**			
	3	**-**	**-**	**-**	**40**	**60**	**3**

Course Objectives

Develop ability to

1. To provide fundamental knowledge of computer graphics systems and devices.
2. To develop understanding of raster graphics and drawing algorithms for basic primitives.
3. To introduce 2D and 3D transformations using matrices and homogeneous coordinates.
4. To explain viewing transformations, clipping techniques, and object representations.
5. To impart knowledge of animation, visible surface detection, and color models.

Course Outcomes

At the end of the course, student would be able to

1. Explain fundamentals of computer graphics systems and applications.
2. Implement drawing and filling algorithms for basic primitives.
3. Apply 2D and 3D geometric transformations.
4. Analyze object representation, viewing, clipping, and rendering techniques.
5. Evaluate animation and visible surface detection methods.

UNIT - I

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random-scan systems, graphics monitors and work stations and input devices.

Output primitives: Points and lines, line drawing algorithms (DDA and Bresenham's Algorithm) circle generating algorithms and ellipse - generating algorithms

Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms

UNIT - II

2-D geometric transformations: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, clipping operations, point clipping, Line clipping-Cohen Sutherland algorithms, Polygon clipping-Sutherland Hodgeman polygon clipping algorithm.

UNIT - III

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, Polygon rendering methods, color models and color applications.

UNIT - IV

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, projections, view volume and general projection transforms and clipping.

UNIT - V

Computer animation: Design of animation sequence, general computer animation functions, raster animations, computer animation languages, key frame systems, motion specifications.

Visible surface detection methods: Classification, back-face detection, depth-buffer method, BSPtree method, area sub-division method and octree method.

Suggested Readings:

1. "Computer Graphics C version", Donald Hearn and M. Pauline Baker, Pearson Education
2. Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.

Course Code	Course Title						Core/Elective
U24IT505	DATA SCIENCE WITH R PROGRAMMING						Professional Elective-II
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Python Programming	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Learn basics of R Programming environment: R language, R- studio and R packages
2. Learn various statistical concepts like linear and logistic regression, cluster analysis, time series forecasting
3. Learn Decision tree induction, association rule mining and text mining
4. Analyze a data set in R and present findings using the appropriate R packages.
5. Prepare or tidy data for in preparation for analysis

Course Outcomes

At the end of the course, student would be able to

1. Identify and execute basic syntax and programs in R.
2. Perform the Matrix operations using R built in functions.
3. Apply non numeric values in vectors
4. Create the list and data frames.
5. Exploit the graph using ggplot2.

UNIT – I

Overview of R: History and Overview of R- Basic Features of R-Design of the R System- Installation of R-Console and Editor Panes, Comments, Installing and Loading R Packages, Help Files and Function Documentation Saving Work and Exiting R-Conventions, R for Basic Math, Arithmetic-Logarithms and Exponentials E-Notation, Assigning Objects, Vectors- Creating a Vector, Sequences, Repetition, Sorting, and Lengths- Sub setting and Element Extraction- Vector-Oriented Behaviour.

UNIT – II

Exploring Data in R: Introduction, Data Frames, R Functions for Understanding Data in Data Frames, Load Data Frames, Exploring Data, Data Summary, Finding the Missing Values, Invalid Values and Outliers, Descriptive Statistics, Spotting Problems in Data with Visualization.

UNIT- III

Linear Regression Using R: Introduction, Model Fitting, Linear Regression, Assumptions of Linear Regression, Validating Linear Assumption. Logistic Regression: Introduction, What Is Regression? Introduction to Generalized Linear Model, Logistic Regression, Binary Logistic Regression, Diagnosing Logistic Regression, Multinomial Logistic Regression Model.

UNIT – IV

Lists And Data Frames: Lists of Objects-Component Access-Naming-Nesting-Data Frames-Adding Data Columns and Combining Data Frames-Logical Record Subsets-Some Special Values-Infinity-NaN-NA-NULL Attributes-Object-Class-Is-Dot Object-Checking Functions-As-Dot Coercion Functions.

UNIT - V

Clustering: Introduction, What Is Clustering, Basic Concepts in Clustering, Hierarchical Clustering, K- Means Algorithm, CURE Algorithm, clustering in Non-Euclidean Space, Clustering for Streams and Parallelism. Association Rules: Introduction, Frequent Itemset, Data Structure Overview, Mining Algorithm Interfaces, Auxiliary Functions, Sampling from Transaction, Generating Synthetic Transaction Data, Additional Measures of Interestingness, Distance Based Clustering Transaction and Association.

Suggested Readings:

1. Nina Zumel, Practical Data Science with R, Manning Publications, 2014.
2. Peter Bruce and Andrew Bruce, Practical Statistics for Data Scientists, O'Reilly, 2017.
3. Hadley Wickham and Garrett Grolemund, R for Data Science, O'Reilly, 2017.
4. Roger D Peng, R Programming for Data science, Lean Publishing, 2016.
5. Rafael A Irizarry, Introduction to Data Science, Lean Publishing,2016.

Course Code	Course Title						Core/Elective
U24IT506	MOBILE APPLICATION DEVELOPMENT						Professional Elective-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Python Programming	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Impart knowledge on Android OS design and Features.
2. Know the Android application components, user interface.
3. Implement data persistence techniques such as shared preferences, file storage, and content providers for mobile apps.
4. Explore mobile networking and telephony services, including background services, multithreading, and system broadcasts.
5. Compare Android and iOS development platforms, understanding the navigation, design, and data handling approaches in iOS.

Course Outcomes

At the end of the course, student would be able to

1. Describe the Android platform, its features, architecture, and tools like ADB and ART.
2. Design basic building blocks of Android programming required for App development.
3. Apply different data persistence schemes for data storage mechanism in Android.
4. Develop background services and handle asynchronous tasks using multithreading and broadcast receivers.
5. Create and deploy location-based services using Maps API and geolocation features.

UNIT-I

Introduction to Android Operating System: Android OS and Features – Android development framework, Installing and running applications on Android Studio, Creating AVDs, Types of Android
Application, Creating Activities, Activity Life Cycle, Activity states, monitoring state changes.

UNIT- II

Android application components: Android Manifest file, Externalizing recourses like Simple Values, Drawables, Layouts, Menus.
Building User Interfaces: Fundamental Android UI design, Layouts – Linear, Relative, Grid and Table Layouts. User Interface (UI) Components.

UNIT-III

Fragments: Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities.

UNIT-IV

Intents and Broadcasts: Using intents to launch Activities, Types of Intents, Passing data to Intents, Getting results from Activities, Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters.

UNIT-V

Database: Introduction to SQLite database, creating and opening a database, creating tables, inserting
retrieving and deleting data. Creating content providers.

Suggested Readings:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012
2. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
3. Android Application Development (with Kitkat Support), Black Book, Pradeep Kothari, 2014, Dreamtech Press publisher, Kogent Learning Inc.,2014.

Course Code	Course Title						Core/Elective
U24IT507	SOFTWARE ENGINEERING						Professional Elective-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming for Problem Solving	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand basic concepts of software development processes of defining a process framework.
2. Impart knowledge on various phases, methodologies and practices of software development.
3. Understand the concepts of quality standards and software change management.
4. Understand the importance of testing in software development and study various testing strategies.
5. Identify the risks and to study different estimation techniques.

Course Outcomes

At the end of the course, student would be able to

1. Define different software development processes and their usability in different problem domains.
2. Explain the process of requirements collection, analyzing, and modeling requirements for effective understanding and communication with stake holders.
3. Design and Develop the architecture of real-world problems towards developing a blue print for implementation.
4. Understand the concepts of software equality, testing and maintenance.
5. Discuss the concepts related to Risk management and Software project Estimation

UNIT-I

Introduction to Software Engineering: A generic view of process, Software Engineering process framework, Nature of Software, Software Myths, Process Models.

Agile Development: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Building the Requirement Model, Negotiating Requirements, Validating Requirements.

Design Concepts: Design within the Context of Software Engineering, the Design Process, Design Concepts, Design Principles, Concept of Structural Design, DFD's, Example Data flow diagrams.

Architectural Design: Software Architecture, Architecture Genres, Architecture Styles, Architecture Designs.

UNIT-III

Software Quality Assurance: Background Issues, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan.

Risk Management: Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.

UNIT-IV

Testing Conventional Applications: Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black–Box Testing.

Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Validation Testing, System Testing, The Art of Debugging.

UNIT-V

Software Configuration Management: Software Configuration Management, The SCM Repository.

Product Metrics: A Frame work for Product Metrics, Metrics for the Requirements Model, Metrics for the Design Model, Metrics for Testing, Metrics for Maintenance.

Estimation: Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Specialized Estimation Techniques.

Software Process Improvement: The SPI Process, The CMMI.

Suggested Readings:

1. Software Engineering: A Practitioners Approach, Roger S.Pressman, McGrawHill, Eighth Edition, 2019.
2. Software Engineering, Ian Somerville, Ian Somerville, Pearson Education, Eighth Edition, 2007.
3. An Integrated Approach to Software Engineering, Pankaj Jalote, Narosa Publishing House, Third Edition 2008.

Course Code	Course Title						Core/Elective
U24IT508	DATA MINING						Professional Elective-I
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DBMS	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand data classification, data preprocessing and data mining applications.
2. Work with the patterns, associations and correlations can be obtained on data.
3. Describe various classification techniques.
4. Understand the classification and clustering techniques can be implemented & perform its evaluation.
5. Learn how complex data mining can be performed.

Course Outcomes

At the end of the course, student would be able to

1. Classify types of data, perform preprocessing of data and appreciate applications of data mining.
2. Analyze data for mining frequent patterns, Associations and Correlations.
3. Perform the classification by using decision tree induction, Bayes classification methods etc. and evaluate the classifier.
4. Select and perform clustering, outlier analysis detection methods.
5. Perform Text mining, Spatial Mining, Web mining and Multimedia mining.

UNIT-I

Introduction: Fundamentals of Data Mining, Kinds of Patterns can be mined, Technologies used, Applications and issues in Data Mining. Types of Data: Attribute types, Basic Statistical Descriptions of Data, Measuring data similarity and Dissimilarity. Data Pre-Processing: Need of Pre-processing, Data Cleaning, Data Integration, Data Reduction, Data Transformation.

UNIT-II

Mining Frequent Patterns, Associations and Correlations: Market Basket Analysis, Association rule mining, frequent item set mining methods, mining various kinds of association rule, Constraint based frequent pattern mining.

UNIT–III

Classification: General approach to classification, Classification by Decision tree induction, Classification by back Propagation, Lazy learners, other classification methods, Prediction, Evaluating the accuracy of classifier, increasing the accuracy of classifier.

UNIT—IV

Cluster Analysis: Basic Clustering methods, Partitioning methods, Density-based methods, Grid-based methods, and Evaluation of clustering, Outlier Analysis and detection methods.

UNIT—V

Mining Complex Data, Applications and Trends: Mining complex data: Spatial mining, Text Mining, Multimedia Mining, Web Mining, Data Mining Applications and Data Mining Trends.

Suggested Readings:

1. Data Mining: Concepts and Techniques, HanJ & Kamber M, Harcourt India, Elsevier India, Second Edition.
2. Introduction to Data Mining, Pang-Ning Tan. Michael Steinback, Vipin Kumar, Pearson Education,2008.
3. Datamining: Introductory and Advanced Topics, Margaret H Dunham, S.Sridhar, Pearson Education,2008.
4. Data Warehousing: Architecture and Implementation, Humphires, Hawkins, Dy. Pearson Education,2009.
5. Data Warehousing in the Real World, Anahory, Murray, Pearson Education,2008.

Course Code	Course Title						Core/Elective
U24IT602	ADVANCED INTERNET OF THINGS						Professional Elective-II
Prerequisite	Hours Per Week				CIE	SEE	Credits
Programming for Problem Solving	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand the basics of IoT and IIOT.
2 Impart necessary and practical knowledge in Industrial Internet of Things.
3.Working on applications and Raspberry Pi.
4 Develop skills required to build real-time IIoT based projects
5.Evaluate the use of Smart Transportation, Water Management learning through case studies

Course Outcomes

At the end of the course, student would be able to

1. Explore the Internet of Things and IIOT basics components.
2. Illustrate working of I/O devices, sensors & communication module.
3. Analyse the use of protocols in IoT.
4. Interface I/O devices, Sensors & communication module
5. Develop real time IoT based projects.

UNIT-I

Internet of Things: The Third Wave? Advantages and Disadvantages of IoT. The Industrial Internet of Things (IIoT): Definition of IIoT, IoT, and M2M, IIoT Challenges, IIoT Requirements, IIoT Benefits. Internet of Things: More than Smart "Things": IoT key attributes, Three Major Challenges Facing IoT: Technology, Technological Challenges, Business, Categories of IoT, Architecture of IoT.

.

UNIT-II

IoT Implementation and Challenges: Components of IoT Implementation: Sensors, Networks, Standards, Intelligent analysis, Intelligent actions. IoT Standardization and Implementation Challenges, Communication modules, I/O interfaces, Programming API's.

UNIT-III

Configuring Raspberry Pi, MicroPython Pyboard, and Jetson Nano for Python: Raspberry Pi Board Feature, Configuration of Raspberry Pi, Simple Applications with Raspberry Pi: OLED Display Interface, Camera Interfacing, Motor Control (DC Motor, Stepper Motor, and Servo Motor), Raspberry Pi and Mobile Interface Through Bluetooth.

UNIT-IV

IoT data protocols: MQTT, CoAP, AMQP, DDS, HTTP, WebSocket. Network Protocols for IoT: 6LowPAN, RPL, WiFi, Bluetooth, ZigBee, Z-Wave, LoRaWan, , XMPP.

UNIT-V

IIoT Case Studies: Smart Grids for Energy Management, Connected Agriculture, Smart Buildings and Facilities Management, Supply Chain Optimization, Connected Healthcare, Smart Retail, Smart Transportation, Water Management

Suggested Reading:

1. Ahmed Banafa by Introduction to Internet of Things (IoT) Published 2023 by River Publishers
2. Jivan S. Parab · Madhusudan Ganuji Lanjewar · Marlon Darius Sequeira · Gourish Naik · Arman Yusuf Shaikh by Python Programming Recipes for IoT Applications , Springer Nature Singapore Pte Ltd. 2023.
3. Arshdeep Bahga, Vijay Madisetti, Internet of Things: A hands on approach, 2014, VPT publishers

Course Code	Course Title						Core/Elective
U24IT603	**GENERATIVE AI**						**Core**
Prerequisite	**Contact Hours per Week**				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
Knowledge on Statistics and Machine Learning	**3**	**-**	**-**	**-**	**40**	**60**	

Course Objectives:

1. To learn the fundamental concepts of Generative AI.
2. To acquire the knowledge of encoders, decoders and autoregressive models.
3. To acquire the knowledge of various generative models for image generation, style transfer and text generation.
4. To learn to apply transforms, prompt engineering and APIs for real world problems.
5. To learn to implement develop application using chat GPTs and open API.

Course Outcomes

1. Understand the fundamental concepts and significance of Generative AI and the unique challenges associated with generative models.
2. Learn the structure, function, and applications of autoencoders and autoregressive models in machine learning.
3. Understand the principles, architecture, and applications of Generative Adversarial Networks for image generation and style transfer.
4. Grasp the architecture and functionality of transformers, and apply prompt engineering techniques using Hugging Face pretrained transformers and APIs.
5. Explore the advancements, capabilities, and practical applications of GPT models, including developing a GPT-3 powered question-answering application.

UNIT-I

Introduction: An Introduction to Generative AI, Applications of AI, The rules of Probability, Why use generative models, Unique challenges of generative models.

UNIT-II

Auto Encoders and Autoregressive Models: Auto encoders, Regularized autoencoders, Stochastic Encoders and Decoders, Autoregressive Models, Fully Visible Sigmoid Belief Network (FVSBN), Neural Autoregressive Density Estimation (NADE), Masked Autoencoder for Distribution Estimation (MADE)

UNIT-III

Generative Adversarial Network: Generative Adversarial Networks, Vanilla GAN, Progressive GAN, Style transfer and Image transformation, Image Generation with GANs, Style Transfer with GANs

UNIT-IV

Transformers and Prompt Engineering: Transformers, Large Language Models, MLM/NSP, Generative Pretrained Transformers (GPT), Task-Specific GPT Fine Tuning, Prompt Engineering, Hugging Face pretrained Transformers, Hugging Face APIs

UNIT-V

Chat GPTs and OpenAI GPT 3, 3.5, 4, OpenAI APIs, working with the OpenAI Playground, Application and Use Cases: Content Filtering, Generating and Transforming Text, Classifying and Categorizing Text, building a GPT-3 Powered Question Answering App

Text Books:

1. Steve Tingiris, Exploring GPT-3, Packt Publishing Ltd. UK, 2021
2. Joseph Babcock, Raghav Bali, Generative AI with Python and TensorFlow 2, Packt Publishing Ltd. UK, 2021

Suggested Reading:

1. Sabit Ekin, Prompt Engineering for ChatGPT: A Quick Guide to Techniques, Tips, and Best Practices, 2023
2. Chris Fregly, Antje Barth, Generative AI on AWS, O'Reilly, 2023
3. Auffarth, Generative AI with LangChain, Packt Publishing, 2023

Web Resources:

1. https://huggingface.co/
2. https://www.udemy.com/course/generative-ai-for-beginners-b/
3. https://www.coursera.org/learn/generative-ai-with-llms
4. https://ai.google/discover/generativeai/

Course Code	Course Title						Core/Elective
U24IT604	INFORMATION SECURITY						Professional Elective-II
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Computer Networks	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Learn legal and technical issues in building secure information systems
2. Provide an understanding of network security
3. Give an exposure to the students towards security standards and practices
4. Provide an understanding of cryptography and Security Tools
5. Learn Implementing Information Security and Information Security Maintenance

Course Outcomes

At the end of the course, student would be able to

1. Describe the steps in Security Systems development life cycle (Sec SDLC).
2. Understand the common threats, legal and ethical issues.
3. Identify security and risk management for business needs & use of security frameworks in preparing security blue print for the organization.
4. Usage of reactive solutions, firewalls, software and Intrusion Detection techniques.
5. Use ethical hacking tools and secure communication protocols and use of the technical and non-technical security aspects.

UNIT – I

Introduction: History, Critical Characteristics of Information, NSTISSC Security Model, Components of an Information System, Securing the Components, Balancing Security and Access, The SDLC, The Security SDLC.
Need for Security: Business Needs, Threats, Attacks, and Secure Software Development

UNIT – II

Legal, Ethical and Professional Issues: Law and ethics in Information Security, Relevant U.S. Laws, International Laws and Legal Bodies, Ethics and Information Security.
Risk Management: Overview, Risk Identification, Risk Assessment, Risk Control Strategies, selecting a Risk Control Strategy, Quantitative versus Qualitative Risk Control Practices, Risk Management Discussion Points, Recommended Risk Control Practices

UNIT- III

Planning for Security: Security policy, Standards and Practices, Security Blue Print, Security Education, Continuity strategies.
Security Technology: Firewalls and VPNs: Physical Design, Firewalls, Protecting Remote connections

UNIT – IV

Security Technology: Intrusion Detection, Access Control, and other Security Tools: Intrusion Detection and Prevention Systems-Scanning, and Analysis Tools- Access Control Devices.
Cryptography: Foundations of Cryptology, Cipher methods, Cryptographic Algorithms, Cryptographic Tools, Protocols for Secure Communications, Attacks on Cryptosystems

UNIT - V

Implementing Information Security: Information security project management, Technical topics of implementation, Non-Technical Aspects of implementation, Security Certification and Accreditation.
Information Security Maintenance: Security management models, Maintenance model Short case studies in Cryptography and Security, Secure Multi party calculation, Virtual Elections, Single Sign On, Secure Inter Branch Payment transactions, Cross site scripting vulnerability.

Suggested Readings:

1. Michael E Whitman and Herbert J Mattord, Principles of Information Security, Cengage Learning, 6 th Edition 2018
2. Atul khate, Cryptography and Network Security" 4 th edition , Tata Mc Graw Hill , 2019

Course Code	Course Title						Core/Elective
U24IT605	SOFTWARE TESTING AND QUALITY ASSURANCE						Professional Elective-II
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Software Engineering	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Demonstration of software quality assurance and testing as a fundamental component of software lifecycle.
2. Define the scope of software projects.
3. Apply software quality assurance and testing activities using modern software tools.
4. Estimate cost of a project and manage budgets and prepare test plans and schedules for a software quality assurance and testing project.
5. Develop software quality assurance and testing project staffing requirements and effectively manage a project.

Course Outcomes

At the end of the course, student would be able to

1.Apply modern software testing processes in relation to software development and project management.
2.Create test strategies and plans, design test cases, prioritize and execute them.
3.Manage incidents and risks within a project.
4.Contribute to efficient delivery of software solutions and implement improvements in the software development processes.
5.Gain expertise in designing, implementation and development of computer based systems and IT processes.

UNIT – I

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

UNIT – II

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

UNIT- III

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

UNIT – IV

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

UNIT - V

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

Suggested Readings:

1. Daniel Galin, Software Quality Assurance–From Theory to Implementation, Pearson Education.2004
2. Mordechai Ben Menachem / Garry S. Marliss, Software Quality–Producing Practical, Consistent Software, BS Publications, 2014
3. William E. Perry, Effective Methods for Software Testing, 2nd Edition, Wiley.

Course Code	Course Title						Core/Elective
U24IT606	CLOUD COMPUTING						Professional Elective-II
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Operating Systems	L	T	D	P			
	3	-	-	-	40	60	3

Course Objectives

Develop ability to

1. Understand basic concepts of cloud computing & enabling Technologies
2. Perform Auto Scaling , capacity planning and load balancing in cloud
3. Understand security, privacy and compliance issues in clouds
4. Introduce cloud management standards and programming models
5. Work with the load balancing techniques

Course Outcomes

At the end of the course, student would be able to

1. Understand the architecture and concept of different cloud models: IaaS, PaaS, SaaS
2. Create virtual Machine images and deploy them on cloud
3. Identify Security and compliance issues in clouds
4. Analyze the various service provides and the security issues with service models.
5. Work with the cloud management and the enterprise applications.

UNIT – I

Computing Paradigms: High-Performance Computing, Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Bio computing, Mobile Computing, Quantum Computing, Optical Computing, Nano computing.

UNIT – II

Introduction, Benefits and challenges, Cloud computing services, Resource Virtualization, Resource pooling sharing and provisioning Scaling in the Cloud, Capacity Planning, Load Balancing, File System and Storage

UNIT- III

Cloud Computing Architecture and Management: Cloud architecture, Layer, Anatomy of the Cloud, Network Connectivity in Cloud Computing, Applications, on the Cloud, Managing the Cloud, Managing the Cloud Infrastructure Managing the Cloud application, Migrating Application to Cloud, Phases of Cloud Migration Approaches for Cloud Migration.

UNIT – IV

Cloud Service Models: Infrastructure as a Service, Characteristics of IaaS. Suitability of IaaS, Pros and Cons of IaaS, Summary of IaaS Providers, Platform as a Service, Characteristics of PaaS, Suitability of PaaS, Pros and Cons of PaaS, Summary of PaaS Providers, Software as a Service, Characteristics of SaaS, Suitability of SaaS, Pros and Cons of SaaS, Summary of SaaS Providers, Other Cloud Service Models.

UNIT - V

Portability and Interoperability Issues, Cloud Management and a Programming Model Case Study, Popular Cloud Services Enterprise architecture and SOA, Enterprise Software, Enterprise Custom Applications, Workflow and Business Processes, Enterprise Analytics and Search, Enterprise Cloud Computing Ecosystem.

Suggested Readings:

1. Essentials of cloud Computing: K. Chandrasekhran, CRC press, 2014
2. Cloud Computing – Sandeep Bhowmik, Cambridge University Press, 2017.
3. Enterprise Cloud Computing – Technology, Architecture, Applications by Gautam Shroff, Cambridge University Press, 2016.