LTIMindree | EduTech | Joint PG Courses

Syllabus for M.Tech. (IOT)

Document Version / Details: Ver. 1.5/21-06-2024

Course Overview

The Internet of Things (IoT) is rapidly changing the world around us, transforming a huge range of physical objects through digital intelligence. Beyond our ubiquitous smart personal devices, IoT is revolutionizing the way companies do business –helping them become faster, smarter, safer, and more efficient.

The number of Internet of Things (IoT) devices worldwide is forecast to almost double from 15.1 billion in 2020 to more than 29 billion IoT devices in 2030. In 2030, the highest number of IoT devices will be found in China with around 8 billion consumer devices.

Top emerging technologies powering IoT includes:

1. Embedded engineering
2. Intelligent applications in Cloud
3. AI augmented development
4. Platform engineering and security

We offer M.Tech in Internet of Things which is a specially designed two-year post-graduate programme for engineers to leverage the advantages of emerging trends in market and get them skilled to do their job as soon as they complete their Post graduate.

We focus on building Edge gateways, IoT middleware, Data engineering and Edge AI domain. This MTech offering is an amalgam of the fundamentals, essentials of IOT and the intricate deep practical knowledge of engineering IOT applications. The curriculum and syllabus are designed and developed by LTIMindtree Industry practitioners, chief architects, and experts who have more than 2 decades of experience in this field with rich technical and domain knowledge, and it will be integrated with a real time lab simulation to help students get industry exposure while learning.

The outcome of this course is to shape a student's career in emerging technologies for designing and implementing an IoT devices and applications end-to-end.

Total programme consists of theory, theory + integrated lab, exclusive laboratory, soft skills, and projects.

Course Structure

Joint PG Course - MTech (IOT)

Semester I							
Subject	**Course Code**	**Hours**	**Type**	**L**	**T**	**P**	**Credits**
Mathematical Foundations		60	C	4	0	0	4
Edge Engineering		60	C	4	0	0	4
Professional Elective - I		45	E	3	0	0	3
Programming for Data Analysis		45	C	3	0	0	3
Applied ML		45	C	3	0	0	3
Professional Skills - I		30	A	0	0	2	1
Programming Lab		60	P	0	0	4	2
Edge engineering Lab		30	P	0	0	2	1
Total		375					21
Professional Elective - I: Java or .Net programming for IOT application							

Semester II							
Subject	**Course Code**	**Hours**	**Type**	**L**	**T**	**P**	**Credits**
IoT Middleware and Data Engineering		45	C	3	0	0	3
Advanced Signal Processing and Data Analysis		60	C	4	0	0	4
Professional Elective – II		45	E	3	0	0	3
Professional Elective - III		45	E	3	0	0	3
Control Systems		45	C	3	0	0	3
Professional Skills - II		30	A	0	0	2	1
Embedded Programming Lab		30	P	0	0	2	1
IoT middleware and Data Engineering Lab		60	P	0	0	4	2
Total		375					20
Professional Elective - II: Application Architecture and Deployment (Mandate) / CS62101							
Professional Elective - III: Embedded Programming (Mandate) / CS60308							

Semester III						
Subject	**Course Code**	**Type**	**L**	**T**	**P (Hours)**	**Credits**
Professional Elective - IV		C	3	0	0	3
Project - I		P	-	-	-	16
Total						19
Professional Elective - IV: IOT industry best practices, standards and compliances						

Semester IV						
Subject	**Course Code**	**Type**	**L**	**T**	**P (Hours)**	**Credits**
Project- II		P	-	-	-	20
Total						20

Semester – I

Title	Mathematical Foundations	Code	
Prerequisite		Credits | Total Hours	4-0-0 [4] | 60

Course Outcome:
Students will be able to refresh the statistical knowledge learnt earlier with hands-on practical expertise.

CO1: Refresh the mathematics knowledge with respect to Linear algebra, Vectors, Projections, Principal Component Analysis and Generative Models
CO2: Refresh the mathematics knowledge with respect to Matrices, Gradient Calculus, Optimization models.
CO3: Refresh the mathematics knowledge with respect to probability, statistics.

Unit 1: Linear Algebra 12

Systems of Linear Equations - Machine learning motivation - A geometric notion of singularity - Singular vs non-singular matrices - Linear dependence and independence - Matrix row-reduction - Row operations that preserve singularity - The rank of a matrix - Row echelon form - Reduced row echelon form- LU decomposition- Solving Systems of Linear Equations - Machine learning motivation - Solving non- singular systems of linear equations - Solving singular systems of linear equations - Solving systems of equations with more variables - Gaussian elimination.

Unit 2: Probability & Statistics 12

Introduction to probability - Concept of probability: repeated random trials - Conditional probability and independence - Random variables - Cumulative distribution function - Discrete random variables: Binomial distribution - Probability mass function - Continuous random variables: Uniform distribution - Continuous random variables: Gaussian distribution -Joint distributions - Marginal and conditional distributions - Independence - covariance - Multivariate normal distribution - Sampling and point estimates - Interval estimation -Confidence intervals – Confidence Interval for mean of population - Biased vs Unbiased estimates-Maximum likelihood estimation - Intuition behind maximum likelihood estimation - Hypothesis testing - Describing samples: sample proportion and sample mean - Two types of errors - Test for proportion and means - Two sample inference for difference between groups.

Unit 3: Bayesian Statistics & its applications in various fields 12

Bayesian statistics and its applications in various fields - Bayesian Learning: Bayes theorem - maximum likelihood and least squared error hypotheses – Naïve Bayes classifier- Bayesian belief networks- gradient ascent training of Bayesian networks- learning the structure of Bayesian networks- the EM algorithm- mixture of models- Markov models- hidden Markov models - Time series analysis and forecasting techniques - Basic Properties of time-series data: Distribution and moments- Stationarity- Autocorrelation- Heteroscedasticity- Normality- Survival Analysis.

Unit 4: Non-Parametric Statistics 12

Non-parametric Statistics - Chi square test- Sign test -Wilcoxon signed rank test - Mann Whitney test - Run test - Kolmogorov Smirnov test - Spearmann and Kendall's test - Tolerance region.

Unit 5: Multivariate Statistical Methods for Analyzing Complex Datasets 12

Multivariate statistical methods for analysing complex datasets - Factor Analysis - Cluster Analysis- Regression Analysis - Discriminant Analysis.

Reference Books:

1. James D. Miller, Statistics for Data Science - By Packt Publishing (2017)
2. IND James D. Hamilton, Time Series Analysis – By Levant Books (2012)

Title	Edge Engineering	**Code**	
Prerequisite		**Credits | Total Hours**	4-0-0 [4] |60

Course Outcomes:
Associates will be able to develop programs that runs on edge device and system level programming that run on edge.

CO1: Learn basic concepts of network types, models and security.
CO2: Learn about different IoT sensors and actuators.
CO3: Understand IoT connectivity and communication technologies, communication protocols and industry standard protocols.
CO4: Program edge devices and develop cloud edge gateways.
CO5: Understand different IoT applications and futuristic trends in market.

Unit 1: Networking 12

Introduction to IoT; Network Types: Connection types - Physical topology - Network reachability; Layered Network Models: OSI Model - Internet protocol suite; TCP/IP Transport layer: Connectionless service - Connection-oriented service; Basics of Network Security: Introduction – Security - Symmetric key cryptography - Asymmetric key cryptography - Message Integrity and Authenticity - Digital signatures - Internet Security - Network layer security - Application layer security- Firewall; Predecessors of IoT: Wireless Sensor Networks - Architectural components of WSN - Machine-to-Machine Communications - Architectural components of M2M- Cyber Physical Systems.

Unit 2: IoT Sensors and processing 12

Emergence of IoT: IoT versus M2M - IoT versus WoT - Enabling IoT and the Complex Interdependence of Technologies - IoT Networking Components - Addressing Strategies in IoT - Address management classes - Addressing during node mobility; IoT Sensing and Actuation: Introduction – Sensors - Sensor Characteristics - Sensorial Deviations- Sensing Types - Scalar sensing - Virtual sensing - Sensing Considerations - Actuator Types - Hydraulic actuators - Pneumatic actuators - Electric actuators - Thermal or magnetic actuators - Mechanical actuators - Soft actuators - Shape memory polymers - Actuator Characteristics; IoT Processing Topologies and Types: Data Format - Structured data - Unstructured data - Importance of Processing in IoT - Processing Topologies - On-site processing - IoT Device Design and Selection Considerations.

Unit 3: IoT Connectivity and Communication 12

IoT Connectivity Technologies: Introduction - IEEE 802.15.4 – Zigbee- Thread – Wireless HART – RFID – NFC - Z-Wave – Weightless – Sigfox – LoRa - NB-IoT - Wi-Fi – Bluetooth; IoT Communication Technologies: Introduction - Constrained nodes - Constrained networks - Types of constrained devices - Low power and lossy networks - Infrastructure Protocols - Internet protocol version 6 (IPv6) - Data Protocols – MQTT - MQTT-SN – CoAP- AMQP – XMPP – REST – WebSocket Device Management; Serial Communication Protocol: UART - I2C – SPI; Industrial Protocols: Modbus – Profibus – BACnet.

Unit 4: Programming Edge and Cloud Edge Gateway 12

Programming Edge Devices: Introduction to C - Introduction to Arduino Programming - Sensors and Actuators integration to Arduino - Introduction to Python Programming - Introduction to RaspberryPi - Sensors and Actuators integration to RaspberryPi - Implementation of IoT on Raspberrypi; Cloud Edge gateway Design & Programming: Azure edge gateway - Design & Implementation - Simulation of device data with program and usage of edge as buffer/local storage - Usage of Azure SQL & blob edge - Deployment of edge in Kubernetes - AKRI and Edge essentials - Scalability and high availability of edge gateway - Project assignment- AWS edge gateway - Design & Implementation- Simulation of device data with program and usage of edge as buffer/local storage - AWS Greengrass & IoT Sitewise - Deployment of edge in Kubernetes - Scalability and high availability of edge gateway - Project assignment - Build Azure/AWS edge connecting to physical device which will be designed in lab or simulated device.

Unit 5: IoT Applications 12

IoT Case Studies & Future Trends: Agricultural IoT - Components of an agricultural IoT - Advantages of IoT in agriculture - Vehicular IoT - Components of vehicular IoT - Advantages of vehicular IoT - Healthcare IoT - Components of healthcare IoT - Advantages and risk of healthcare IoT.

Reading Materials / References:

1. AK Dubey, Vijayan Sugumaran, Peter Han Joo Chong (2023), Advanced IoT Sensors, Networks and Systems, Springer.
2. C. Santhiya and S. Padmavathi (2023), Perspective Chapter: A View – Cloud-Edge Computing Technology
3. Gerardus Blokdyk (2008), IoT Communication Protocols Second Edition, 5STARCooks.

Title	Java programming for IOT applications (Professional Elective - I)	**Code**	
Prerequisite		**Credits | Total Hours**	3-0-0 [3] |45

Course Outcomes:

Associates will be able to develop programs that runs on Java.

CO1: Learn basic java programming.
CO2: Understand OOPS concept in-depth.
CO3: Develop advanced java using collections, write code to handle concurrency, interfaces, network programming, connecting to database using JDBC and regular expressions.

Unit 1: Java Basics 9

Introduction to Java language: Brief history of Java -Java as a language - JDK, JRE and JVM - How does - JRE work - Java virtual machine; Datatypes and variables: Data types and variables - Non-primitive string - Primitive Data types - Implicit type casting or widening - Explicit type casting or narrowing; Operators: Operators - Arithmetic Operators - Unary Operators - Bitwise Logic Operators – Bitwise -Shift Operators - Comparison Operators - Logic operators - Assignment Operators -Ternary or Conditional Operator - Operator Precedence; Conditional flow control: If and Nested If statement

- Switch case statement; Loops: For Loop - While Loop - Do While Loop - Infinite Loop - Break continue statements - Nested Loop - Labeled loop; Arrays: Arrays; Practice: Problem statement and solution; Methods: Creating Methods - Calling Methods - Method overloading - Recursive Methods.

Unit 2: OOPS in Java 9

Basics of object-oriented programming: Class, Object – Constructors - Pass by reference, Pass by value - Static fields, class variables, instance variable - static methods or class methods - Static blocks.- Inheritance - Interfaces and implementation; Java packages: Packages - creating user defined packages; Encapsulation: Encapsulation; Inheritance: Inheritance; Polymorphism: Polymorphism;Abstraction: Abstraction; Composition: Composition.

Unit 3: Advanced concepts – I 9

Nested classes: Nested classes - Inner classes - Shadowing - Static Nested classes; Enum types: Enum types; Java string class: Java string class; Wrapper class: Wrapper class.
Java Mathematics on BigInteger, Big Decimal class: Java Mathematics on BigInteger, BigDecimal class.
Java Date and Time: Java Date and Time; Formatting - Decimal, Date & Time, custom Date time formatting: Formatting - Decimal, Date & Time, custom Date time formatting; Arrays utility class: Arrays utility class; Exception handling: Exception handling; Generics: Generics; Collections: List interface – Array List – LinkedList - Vector, Stack Class - Sorting Lists - Hashset class, LinkedHashSet class – TreeSet - Queue interface - Map, HashMap, TreeMap - Immutable collections and Collection factories; Concurrency: Process and Thread Overview -Creating Threads, Pausing Thread -Thread Joins - Inter-thread communication – Deadlock – Livelock; Practice: Practice session.

Unit 4: Advanced concepts - II 9

Lambda expressions; Functional Interfaces: Functional Interfaces; Streams and Generating streams: Streams and Generating streams; I/O Operations and files: I/O Operations and files; Serialization: Serialization; Network programming: Network programming; Java NIO:Java NIO; JDBC Database connectivity: JDBC Database connectivity; Garbage collector and Java Runtime: Garbage collector and Java Runtime; Reflection: Reflection; Annotations: Annotations; Regular expressions: Regular expressions.

Unit 5: Code with us 9

Project for submission: Project.

Reading Materials / References:

1. Kathy Sierra & Bert Bates (2003), Head First Java, Shroff/O'Reilly.
2. Joshua Bloch (2017), Effective Java, Addison-Wesley Professional.
3. D. S. Guru, K. S. Manjunatha, and M. T. Somashekara (2017) Object Oriented Programming with Java, PHI Learning.

Reference Links:

1. https://www.udemy.com/course/java-from-scratch-to-master/? referralCode= 7BD816C575445B169601&gclid=EAIaIQobChMIwfnw3cb5_gIVyWN9Ch08YA3VEAAYASAAEgKsG_D_BwE

Title	.Net programming for IOT applications (Professional Elective - I)	Code	
Prerequisite		Credits |Total Hours	3-0-0 [3] |45

Course Outcomes:

Associates will be able to develop programs on .NET programming.

CO1: Understand basics of .NET programming and SQL server.
CO2: Learn APIs and develop APIs to get data from SQL using stored procedures.
CO3: Dive deep into SOLID principles
CO4: Develop .NET web application and deploy in Azure
CO5: Design and implement real time project in .NET, SQL and deploy in Azure

Unit 1: Introduction to .NET and SQL 9

Introduction to .NET Programming: Basics of .NET - Installation of .NET in windows/mac/Linux; Basics of C#.NET: Console app creation-Variables-Data structures- Operators and conditionals-Conditional statements- Loops, Methods, scope - Practice - problem statements and solution; Intermediate concepts in C#.NET: Models - String operations – Namespaces -Connecting with Databases and database connections -Dapper - Entity framework – Config - File operations - Read and write – JSON -
Model mapping; SQL Basics & Intermediate concepts: SQL Basics - SQL Intermediate - Practice - problem statements and solution.

Unit 2: API Basics and Advanced concepts 9

API Basics: Basics of API development -what is startup.cs - First custom controller - API set up.
-User Models and User Controller -Put & Post -DTOs -Namespaces - EF Setup, User controller and Automapper - Assignment and solution; API Intermediate: User Repository, Auth Table- Login and Registration with details- JWT token creation and validation- Helper Classes-Posts model and controller; Stored procedures: Stored procedure creation- Parameters -Nullable parameters-Temp table- User upsert and delete- Posts Get, Upsert and Delete; API Advanced: API Advanced concepts
-Registration set up end to end- Controller Dynamic parameters- Advanced - assignment and solution-Reusable SQL.

Unit 3: SOLID principles 9

SOLID Principles for C# Developers: Single Responsibility Principle- Open/Closed Principle
-Liskov substitution Principle- Interface segregation Principle- Dependency inversion Principle.

Unit 4: .NET in Azure 9

Azure deployment: Azure App Service -Azure SQL - Deploy .NET apps in Azure.

Unit 5: Code with us 9

Project for submission: Assignment1 and solution -Assignment2 and solution- Project - Problem statement and solution in Azure

Reading Materials / References:

1. Thuan L. Thai, Hoang Lam (2003), .NET Framework Essentials, 3rd Edition, O'Reilly Media, Inc.
2. Pallavi Agarwal (2021), API Fundamentals: An Easy Hands-on Workbook for Beginners, independently published.
3. Anuraj Parameswaran, Tamir Al Balkhi (2023), A Developer's Guide to .NET in Azure, Packt Publishing Limited.

Reference Links:

1. https://www.udemy.com/course/net-core-with-ms-sql-beginner-to-expert/
2. https://www.pluralsight.com/courses/csharp-solid-principles

Title	Programming for Data Analysis	**Code**	
Prerequisite		**Credits |Total Hours**	3-0-0 [3] |45

Course Outcomes:
Associates will be able to develop programs that runs on Pyspark.

CO1: Understand concepts of big data
CO2: Learn spark code in depth on RDD, spark internals, Data frames, visualizations and usage of Spark UI.
CO3: Develop Pyspark programs using RDD, data frames and perform data analysis for structured, unstructured and semi-structed data sources.

Unit 1: Introduction to Big data 9

Definition of Big data - Relational databases - Structured, Unstructured and semi-structured data - Big data tooling and technology - Scale up vs Scale out.

Unit 2: NoSQL and Processing Tools 9

NoSQL, Key-value stores, column-oriented stores - Document stores, Graph stores - Basics of MapReduce - What is Spark, Spark batch, Spark Streaming, Storm - Supportive Operational tools.

Unit 3: Spark core in depth 9

Big data ecosystem- RDD (Resilient Distributed Datasets) - Resource Managers - YARN, standalone mode - Deep dive - Spark internals - Accumulators, Broadcast variables - Spark streaming - Spark SQL and Data frames - Machine Learning (NLP, k-means clustering, PageRank, Shortest Path) – Graph Frames - Visualizations (Matplotlib, Google Charts, D3.js) - Advanced Performance Tuning and Debugging- Spark UI.

Unit 4: PySpark - Programming 9

PySpark coding using RDD - PySpark coding using Data Frames - Semi structured file data analysis - Structured file data analysis - Unstructured file data analysis - Distributed processing challenges - Spark performance tuning - Lazy evaluation - Spark internal execution - step by step - Spark SQL using JDBC

Unit 5: Code with us 9

Project.

Reading Materials / References:

1. Tomasz Drabas, Denny Lee (2017) , Learning PySpark, Packt Publishing.
2. Pedro Duarte Faria (2024), Introduction to PySpark, independently published.
3. Matei Zaharia, Bill Chambers (2018), Spark: The Definitive Guide - Big Data Processing Made Simple (Greyscale Indian Edition), Shroff/O'Reilly.

Reference Link:

1. Introduction to Bigdata, NoSQL and processing tools
2. https://www.udemy.com/course/the-definitive-intro-to-big-data-science/
3. https://www.youtube.com/watch?v=7ooZ4S7Ay6Y&list=PLenbFOGbRkGraxcMaoYLSYMSANr47NMcW
4. https://www.youtube.com/watch?v=vlVnSpJ6TDE&t=1283s
5. https://www.udemy.com/course/best-hands-on-big-data-practices-and-use-cases-using-pyspark/

Title	Applied Machine Learning	**Code**	
Prerequisite		**Credits |Total Hours**	3-0-0 [3] | 45

Course Outcome:
Students will be able to get the knowledge about all the tools and techniques you need to apply machine learning to solve business problems.

CO1: To know about Supervised Learning, Support Vector Machines, Unsupervised Learning
CO2: Get the knowledge about Feature Engineering, Statistical Data Analysis, Outlier Analysis and Detection
CO3: Learn about ML Model Development, Model Evaluation Techniques, Model Deployment and Inferences, Model Explainability

Unit 1: Supervised Learning 15

Implement and understand the cost function and gradient descent for multiple linear regression - Implement and understand methods for improving machine learning models by choosing the learning rate - plotting the learning curve - performing feature engineering - applying polynomial regression - Implement and understand the logistic regression model for classification -Learn why logistic regression is better suited for classification tasks than the linear regression model is - Implement and understand the cost function and gradient descent for logistic regression - Understand the problem of - overfitting - improve model performance using regularization - Implement regularization to improve both regression and classification models

Unit 2: Advanced Learning Algos **15**

Build a neural network for binary classification of handwritten digits using TensorFlow - Gain a deeper understanding by implementing a neural network in Python from scratch - Optionally learn how neural network computations are vectorized to use parallel processing for faster training and prediction - Build a neural network to perform multi-class classification of handwritten digits in TensorFlow -using categorical cross-entropy loss functions and the SoftMax activation - Learn where to use different activation functions – ReLu - linear - sigmoid - SoftMax in a neural network - depending on the task you want your model to perform - Use the advanced Adam optimizer to train your model more efficiently - Discover the value of separating your data set into training - cross-validation -test sets - Choose from various versions of your model using a cross-validation dataset -evaluate its ability to generalize to real-world data using a test dataset - Use learning curves to determine if your model is experiencing high bias or high variance - learn which techniques to apply regularization - adding more data - adding or removing input features to improve your model's performance - Learn how the bias-variance trade-off is different in the age of deep learning - and apply Andrew Ng's advice for handling bias and variance when training neural networks - Learn to apply the iterative loop of machine learning development to train - evaluate - tune your model - Apply data-centric AI to not only tune your model but tune your data using data synthesis or data augmentation to improve your model's performance - Build decision trees and tree ensembles - such as random forest and XGBoost - boosted trees - to make predictions - Learn when to use neural network or tree ensemble models for your task - as these are the two most commonly used supervised learning models in practice today.

Unit 3: Unsupervised Learning and Recommender Systems **15**

Use unsupervised learning techniques for unsupervised learning: including clustering and anomaly detection - Build recommender systems with a collaborative filtering approach and a content-based deep learning method - Build a deep reinforcement learning model - Implement K-mean clustering - Implement anomaly detection - Learn how to choose between supervised learning or anomaly detection to solve certain tasks - Build a recommender system using collaborative filtering - Build a recommender system using a content-based deep learning method - Build a deep reinforcement learning model (Deep Q Network)." - Histograms - Box Plots etc - use of frequency distributions – mean comparisons - cross tabulation - statistical inferences using chi square - t-test and ANOVA - Outlier Analysis and Detection - outlier analysis - density based and distance based.

Reference Books:

1. Hang Li, Machine Learning Methods - By Springer Nature Singapore (2023)

2. Dr. R. Nageswara Rao, Machine Learning in Data Science Using Python - By Dreamtech Press (2022)

Title	Professional skills - I	Code	
Prerequisite		Credits | Total Hours	1-0-0 [1] |30
Course Outcomes:			

- To help the students understand and implement positive outlook, interpret the body language of team members and stakeholders, better interpersonal relationships. Develop into self-motivated professionals with confidence. Practice Responding instead of Reacting
- Create good Presentation and Present with confidence. Also, recognize and manage Stress , Prioritize and Plan.
- To listen to understand. To be able to ask good questions.
- To understand to be a good Team player, Team Dynamics and to understand the Business Ethics
- To be able to write and speak correctly, forming grammatically correct sentences.

Unit 1: Positive Attitude 6

Attitude- Campus to Corporate attitude change, Recognising Negative Attitude, Campus to Corporate attitude change; Attitude at work- Impact of Negative Attitude in the Workplace, Overcoming Negative Attitude, positive attitude, thought process, Building self confidence and Assertiveness;Toxic positivity ;3Es, Motivation-Intrinsic and Extrinsic Motivation, Inspiration vs motivation;Emotional Intelligence-Intro to EI , Four clusters.Transactional Analysis (TA) , SWOT analysis - Professional analysis

Unit 2 : Body Language 6

Importance of Body Language,Five C's of Body Language
Body language in different cultures,Positive Body Language;Voice Control- Pace. Pause and Pitch ;Culture -Inclusivity and Proxemics across Global Cultures, Understanding POSH
Stress Management-What is Stress , Eustress, Reasons of stress (work/ personal);Stress Management Techniques

Unit 3: Presentation Skills 6

Self introduction – Exercises, Why Give Presentations;Craft your message-Plan the visuals,Manage the Response;
How to create an effective presentation -Virtual & Physical,Do's & Don'ts of Presentation Skills,Objection handling,Stage Fear – Causes and Cure,Practice the Delivery
Time Management-Common Time & Energy Wasters, Planning & Prioritizing Time Matrix & Analysis

Unit 4 : Listening & Questioning skills 6

Barriers to effective listening - how to overcome them ;Exercies - Customer Call Flow – Role-play ,Cust calls amongst the team; How to frame Questions, Different kinds of questions,Asking appropriate questions;
Spoken English-Introduction to Parts of Speech and it's usage; Subject- Verb Agreement; Basic conversation skills-sentence construction -SVO

Unit 5 : Team Work 6

Team Work and Ethics-1. Definition of TEAM Team vs Groups. Difference b/w Healthy competition and cut throat competetion,2. Importance of working in teams, Evolution of a TEAM, Benefits of team work;Virtual teams-Challenges and ways to overcome it, Diversity and Inclusion in a team; Development of Teams Stages of team development; Team dynamics-its importance & Interpersonal Skills Development Ethics- to enable students to identify and deal with ethical problems, develop their moral intuitions, which are implicit in everyday choices and actions.
Conflict Management:

Team building Activities- Predetermined/ Predesigned Indoor/ Outdoor activities to build a team, enhance language and inter personal skills

Programming Lab

Experiment No	Topic	Experiment Details	Type
Experiment 1	Java	Implement Library management system utilizing all features of Java	Individual
Experiment 2	Java	Implement OOPs concepts	Individual
Experiment 3	Java	Implement Generics, exception handling	Individual
Experiment 4	Java	Implement file handling I/O and serialization	Individual
Experiment 5	Java	Implement database connectivity using JDBC with MySQL	Individual
Experiment 6	Java	Implement APIs using all the concepts learnt	Individual
Experiment 7	Java	Deploy Java pplication to Azure using Azure DevOps	Individual
Experiment 8	.Net	Implement Library management system utilizing all features of .NET	Individual
Experiment 9	.Net	Implement OOPs concepts	Individual
Experiment 10	.Net	Implement Generics, exception handling	Individual
Experiment 11	.Net	Implement file handling I/O and serialization	Individual
Experiment 12	.Net	Implement database connectivity using JDBC with MySQL	Individual
Experiment 13	.Net	Implement APIs using all the concepts learnt	Individual
Experiment 14	.Net	Deploy the .Net application to Azure using Azure DevOps to Azure App service	Individual
Experiment 15	Programming for Data Analysis	Implement data transformations in PySpark with Dataframes using different spark methods and functions	Individual
Experiment 16	Programming for Data Analysis	Use semistructured dataset stored in Data lake either in json/parquet format, implement and showcase data analysis using pyspark scripts	Individual
Experiment 17	Programming for Data Analysis	Create visualization using Matplotlib for the semi structured dataset	Individual
Experiment 18	Programming for Data Analysis	Implement simple algorithm with publicly available data set and spark ML libraries, visualize and show the results	Individual

Experiment 19	Programming for Data Analysis	Implement real time analytics using spark streaming libraries	Individual

Edge engineering Lab

Experiment No	Topic	Experiment Details	Type
Experiment 1	Edge	Consider raspberry Pi and write python language to capture the real time data from sensors and send it to AWS IoT Core	Individual
Experiment 2	Edge	Consider raspberry Pi and write python language to capture the real time data from sensors and send it to Azure IoT hub	Individual
Experiment 3	Edge	Create asset utilization dashboard in Power BI with real time data simulated from sensors and generate actions with cloud specific services	Individual
Experiment 4	Edge	Create azure edge by installing azure edge runtime in Azure VM, write python simulator to generate device data and use Grafana dashboard to create real time monitoring insights	Individual
Experiment 5	Edge	Create AWS green grass edge run time and use aws iot sitewise edge to transfer the simulated device data to AWS IoT Core and generate real time insights using grafana dashboard	Individual
Experiment 6	Edge	Create an application to implement protocol translation in edge.	Individual
Experiment 7	Edge	Simulate device data with python program in edge VM and implement local storage in edge using either Azure/AWS Services	Individual
Experiment 8	Edge	Implement Smart lightning use case with sensor data transferred from edge in Azure to Azure IoT hub and creation of grafana dashboards for real time insights with 10 KPIs. Find the right sensor for smart lightning and build small model, showcase working solution by designing edge in Azure and real time insight generation using Azure IoT hub, ADLS gen2 and PowerBi/Grafana.	Individual

Semester - II

Title	IoT Middleware, Data engineering	Code	
Prerequisite		Credits | Total Hours	3-0-0 [3] |45

Course Outcomes:
Associates will be able to develop programs that runs Cloud (Azure/AWS) and create visualization with data engineering concepts.

CO1: Learn in-depth concepts of Azure IoT services.
CO2: Learn in-depth concepts of AWS IoT services.
CO3: Understand data engineering and real time stream processing, security and governance.
CO4: Visualize the data at real time to make efficient decisions.
CO5: Perform real time project in Azure and AWS

Unit 1: Azure IoT middleware 9

Azure IoT Middleware - Design & Implementation - Azure IoT hub - Azure function - Azure Kubernetes services - API implementation - Azure App Service - Azure DevOps - deploying IoT apps to Azure - Azure monitor, security & Governance services - Project assignment.

Unit 2: AWS IoT middleware 9

AWS IoT Middleware - Design & Implementation - AWS IoT Core - AWS Lambda functions - microservices implementation - AWS Elastic Kubernetes services - API implementation - DevOps - deploying IoT apps in AWS - Monitoring, security & Governance services - Real time assignment _ Case study to simulate data from edge and build Azure IOT middleware platform to show real time assets and build insights to derive business decisions.

Unit 3: IoT Data Engineering 9

Data engineering - Design & Implementation - Data acquisition -Data Processing & Transformation – PySpark - Real time stream analysis & processing - Data Storage- Monitoring, security & Governance.

Unit 4: Real Time insights generation 9

Data Visualization - Visualization - Brief overview - Real time dashboards - Visualization - Hands on.

Unit 5: Code with us 9

Real time Project - code with us.

Reading Materials / References:

1. Honbo Zhou (2012), The internet of things in the cloud a middleware perspective, CRC Press Inc; 1st edition (29 October 2012).
2. Gerardus Blokdyk (2018), AWS IoT Amazon The Ultimate Step-By-Step Guide, 5STARCooks.
3. Joe Reis and Matt Housley (2022), Fundamentals of Data Engineering: Plan and Build Robust Data Systems (Grayscale Indian Edition), Shroff/O'Reilly.

Title	Application Architecture & Deployment (Professional Elective – III)	**Code**	
Prerequisite		**Credits | Total Hours**	3-0-0 [3] | 45

Course Outcome:
Students to understand how architect and AI Application deployment with important aspects to be taken care of.

CO1: Understand the differences between monolithic and microservices architecture and their respective advantages and disadvantages in AI applications.
CO2: Understand the basics of Kubernetes and how it can be used to manage and deploy AI models in a production environment.
CO3: Understand application programming interfaces (APIs) and their role in integrating AI models into larger systems.
CO4: Understand MLOps and how it can be used to streamline the machine learning lifecycle, from data preparation to model deployment and monitoring.

Unit 1: Monolithic vs Microservices 9

Introduction to Software Architecture and its types - What is Monolithic Architecture and its Importance - Characteristics of Monotithic Architecture - Limitations of Monolithic Architecture - What are Microservices - Working of Microservices - Main Components of Microservices Architecture - Advantages of Microservices - Monolithic vs Microservices - Real World Example of Microservices - Challenges in Microservices.

Unit 2: Application Programming Interface 9

What is an API - How do an API Work - WEB APIs - LOCAL APIs - PROGRAM APIs - SOAP, REST API - What are REST APIs - HTTP methods (GET, POST, PUT, DELETE) - Status Codes and URI structure - SOAP vs REST - What is API testing - Types of Testing - Tools for API Testing - Authentication Mechanisms - Authorization Mechanisms - Role Based Access Control (RBAC)

Unit 3: Containers - An Introduction 9

What is Virtualization - Virtualization in Cloud Computing - Introduction to containerization - Container Lifecycle - Virtualization vs Containerization - Container Security - Serverless Containers - Introduction to Docker - Docker Architecture - Components of Docker - Concept of Docker Images - Docker Commands - Advantages of Docker - Introduction to Orchestration tools

Unit 4: Kubernetes - An Introduction 9

What is Kubernetes (K8s) - Why Kubernetes and not only docker - Kubernetes Components - Node - Control Plane - Networking in Kubernetes - Kubernetes Resources - Pod, Deployment, Service, Volume, Namespace, node, cluster - Storage - Security - Monitoring, Logging, Scaling - Writing YAML files.

Unit 5: ML Operations 9

Introduction to ML Operations - What is SDLC - Stages of SDLC - Waterfall Model - Agile Model - Iterative Model - Importance of Each Models - Model Training - Model Deployment.

Reading Materials:

1. Scott Surovich & Marc Boorshtein, Kubernetes and Docker – By Packt Publishing (2021)

2. Mark Treveil, Nicolas Omont & Clément Stenac, Introducing MLOps: How to Scale Machine Learning in the Enterprise (Grayscale Indian Edition) – By Shroff/O'Reilly (2020)

Title	Advanced signal processing and Data Analysis	Code	
Prerequisite		Credits | Total Hours	4-0-0 [4] |60

Course Outcomes:

CO1: Analyse the effect of sampling and quantisation of signals and appraise its relevance with reference to applications.
CO2: Formulate various transform domain representations of 1D and 2D signals and demonstrate their applications with reference to practical signals.
CO3: Examine finite word length effects and design practical filters for real life applications.
CO4: Demonstrate the effect of sampling rate converters and design distortion free digital filter banks illustrating their applications to process real life signals.
CO5: Analyse and choose architectures to efficiently implement the DSP systems for various applications taking into consideration the practical aspects.
CO6: Develop DSP applications using professional tools.

Unit 1: Analysis of Discrete Time Signals 12

Basic elements of a DSP System – Review of Sampling and Quantisation – Sampling theorem for low pass and band pass signals, uniform and non-uniform quantization, Application of quantisation in lossy compression of signals – Lloyd Max quantizer; Fourier analysis of Continuous and Discrete time signals –Review of Fourier series and Fourier transform, Discrete Time Fourier Transform (DTFT), Discrete Fourier Transform (DFT), Interpretation of DFT Spectrum, Review of DFT properties – Convolution and correlation, Convolution of long sequences, Leakage effect, Windowing – Introduction to other transforms : Discrete Cosine Transform (DCT), Walsh Hadamard Transform (WHT), Karhunen Loeve Transform (KLT) – Applications.

Unit 2: Digital Filters and Implementation 11

Review of FIR and IIR filter design – Notch filter– Comb filter– All pass filters– Applications– Structures for digital filter realization: Signal flow graph and block diagram representations, FIR and IIR Filter structures, Lattice structures – Finite word length effects – Fixed-point and floating-point DSP arithmetic, Effects of quantization, Scaling, Limit cycles in fixed point realizations of IIR digital filters, Limit cycles due to overflow. Quantization effect in DFT and FFT computation.

Unit 3: Multirate Signals and Systems 11

Introduction to multirate signal processing with applications, Multirate System Fundamentals – Decimation and Interpolation, transform domain analysis of Decimators and Interpolators, Decimation and Interpolation filters, Fractional sampling rate alteration, Practical sampling rate converter design, Polyphase decomposition, and efficient structures – Introduction to digital filter banks – The DFT filter bank, Two Channel Quadrature Mirror Filter bank (QMF), Perfect Reconstruction.

Unit 4: Introduction to 2-D Signals and Systems 11

Elementary 2D signals – Linear shift Invariant systems – Separability – 2D convolution – Introduction to 2D transforms :2D DFT, 2D DCT, Application

Reading Materials / References:

1. V.V.S.S.S. Chakravarthy, Vikrant Bhateja, Wendy Flores Fuentes, Jaume Anguera, K. Padma Vasavi, (2023), Advances in Signal Processing, Embedded Systems and IoT: Proceedings of Seventh ICMEET- 2022, Springer.
2. Jae S. Lim (1989), Two-Dimensional Signal and Image Processing, Prentice Hall.
3. P. Vaidyanathanm (1993), Multirate Systems And Filter Banks, Prentice Hall.

Title	Embedded Programming (Professional Elective - III)	Code	
Prerequisite		Credits | Total Hours	3-0-0 [3] |45

Course Outcomes:

Associates will be able to develop programs that runs on edge device and system level programming.

CO1: Develop programs using Basic C concepts.
CO2: Develop programs using Advanced C concepts like Arrays, functions.
CO3: Develop concepts using pointers.
CO3: Handle files in programs
CO4: Create structures, dynamic structures and learn advanced concepts on linked list.

Unit 1: Introduction to programming languages and Basics of C programming 9

Introduction; Programming languages - Introduction- Machine languages (first generation) - Assembly languages (second generation) - High-level languages - Processing a high-level language program - Phase 1 creating a program - Phase 2 translate the source code into machine language- Phase 3 linking - Phase 4 execution; Program design: Algorithms - Structured program development – Documentation - Program; Programming in C: an introduction : A first program - Comments- #include - int main(void){ } - Variable definitions - Assignment statements; Basic concepts of C programming: Indentation- Identifiers - Variables – Concept- Variable definition and data types - Variable initialization - Example: variables - Expressions - Arithmetic expressions - Conditional expressions - Precedence - Assignment statements - The assignment operator - Arithmetic assignment operators - Typecasting - Simple input and output - printf() - scanf() - gets() - getchar()
- Exercises.

Unit 2: Advanced concepts of C 9

Controlling the program flow: Flowchart - Selection- Arraynts - The if selection statement - The if else selection statement - The switch statement - Repetition statements - The for statement - The while statement - The do while statement - break and continue - Loop examples; Functions: Standard functions - Mathematical standard functions - Other standard functions - Generation of random numbers - Programmer-defined functions - Void functions without parameters - Void functions with parameters - Functions with return value - Storage classes and scope of variables - Storage class auto local variables - Storage class extern global variables - Storage class register - Storage class static - Structured programming example; Arrays: Definition - Array declaration - Array initialization - Array usage - Operations on arrays - Passing arrays to functions - Array boundaries - Programming examples using arrays - The sieve of Eratosthenes - Merging arrays; Strings: String constant - String variable - Passing strings to functions - String functions – strlen - strcpy - strcmp - Programming examples using strings - Demonstration of several string functions - Sorting strings alphabetically.

Unit 3: Arrays & Pointers 9

Multidimensional arrays: Two dimensional arrays of numbers – Declaration - Initialization - Matrix usage - Passing a 2D array to a function - 2D array example: Pascal's triangle - Arrays of strings; Sorting and searching arrays: Sorting arrays of numbers - Sorting arrays of strings - Binary search; Pointers: Definition - Declaration and initialization – Declaration - Initialization - Address and dereference operator - Passing arguments to functions - Pass by value - Pass by reference - Pointers and arrays - Pointer versions of some string functions - strlen - strcpy – strcmp - Pointers to functions - Function pointers - Array of function pointers - Function pointers as function argument.

Unit 4: Preprocessors and File handling 9

The comma operator - typedef - Type qualifiers - The enumeration type - Bit operations - Bitwise AND - Bitwise OR - Bitwise OR - Bitwise XOR - One's complement - Left shift - Right shift - Example – Masking; The C preprocessor: The C preprocessor - #define preprocessor directive - Symbolic constants – Macros - #include preprocessor directive - Conditional compilation - #ifdef preprocessor directive - #if preprocessor directive; File handling in C: File pointer - Opening and closing a text file - Read and write symbol to a text file - Read one symbol: fgetc - Write one symbol: fputc - Read and write a full line to a text - Read a full line: fgets - Write a full line: fputs - Formatted read and write to a text file - Formatted printing to a file: fprintf - Formatted reading from a file: fscanf – stdin.

Unit 5: Structures 9

Defining a structure - Accessing structure members - Nested structures - Structures and functions - Comparing structures - Pointers to structures - Files of structures;
Dynamic data structures: Introduction - Linked lists - Definition - Creating a single-linked list - Insertion of a new node in a single-linked list - Removal of a node in a single-linked list - Double-linked list - Circular linked list - Stack – Queue.

Reading Materials / References:

1. Ir. Sofie Beerens (3 January 2000), C for Embedded Systems, CRC Press.
2. E.A. LeeandS.A. Seshia, (2017), Introduction to EmbeddedSystems -ACyber-Physical Systems Approach, Second Edition, MIT Press.
3. Thomas Mailund (2021), Pointers in C Programming: A Modern Approach to Memory Management, Recursive Data Structures, Strings, and Arrays, Apress.

Title	Control systems	Code	
Prerequisite		Credits | Total Hours	3-0-0 [3] |45

Course Outcomes:

Associates will be able to understand advanced concepts of control systems

CO1: Understand basics of control systems and their components
CO2: Learn time response analysis and measure performance. Prepare analytical design for PD, PI, PID control systems
CO3: Learn to analyse frequency response, design plots using Bode plots
CO4: Learn the concepts of stability analysis and analyse the control systems with state variable methods.

Unit 1: Systems Components and their representation 9

Control System: Terminology and Basic Structure-Feed forward and Feedback control theory. Electrical and Mechanical Transfer Function Models-Block diagram Models-Signal flow graphs

Unit 2: Time Response Analysis 9

Transient response-steady state response-Measures of performance of the standard first order and second order system-effect on an additional zero and an additional pole-steady error constant and system- type number-PID control-Analytical design for PD, PI, PID control systems

Unit 3: Frequency response and systems analysis 9

Closed loop frequency response-Performance specification in frequency domain-Frequency response of standard second order system- Bode Plot - Polar Plot-Design of compensators using Bode plots- Cascade lead, lag, and lag-lead compensation.

Unit 4: Concepts of Stability Analysis 9

Concept of stability-Bounded - Input Bounded - Output stability-Routh stability criterion - Relative stability-Root locus concept-Guidelines for sketching root locus-Nyquist stability criterion.

Unit 5: Control system Analysis using state variable methods 9

State variable representation-Conversion of state variable models to transfer functions. Conversion of transfer functions to state variable models-Solution of state equations-Concepts of Controllability and Observability

Reading Materials / References:

1. . Norman S. Nise (2014), Control Systems Engineering, CBS Publishers
2. Katsuhiko Ogata (2015), Modern Control Engineering, Pearson Education India
3. Jing Sun (2018), Control Engineering Fundamentals, De Gruyter.

Title	Professional skills - II	Code	
Prerequisite		Credits | Total Hours	1-0-0 [1] |30

Course Outcome:
Students to understand day in day out terms used in customer environment and demonstrate customer centric approach and practically experience the and important aspects of it.

CO1: To understand what is spoken without distortion and respond appropriately.
CO2: To behave professionally.
CO3: To participate productively in an official meeting keeping etiquette in mind.
CO4: To communicate effectively through writing.
CO5: To behave appropriately in an official environment.
CO6: To be comfortable to dine with colleagues, clients, and leaders comfortably in a formal or informal setting.

Unit 1: Accent Neutralization 6

Identifying and dealing with Mother Tongue Influence (MTI) – Pronunciation - Vowel Sounds and Consonant Sounds – Inflection – Pausing - Reducing rate of speech - Volume and tone – Pitch – Clarity - and enunciation.

Unit 2: Customer Service 6

Customer Service - Different types of customers - Difference between customer service and customer experience - Telephone Etiquette - Handling difficult customers.

Unit 3: Problem Solving and Decision Making 6

Define a Problem - Define Decision Making- Blocks in problem solving - Stereotyping and unconscious biases - The process of Problem Solving and decision making - Problem Analysis- Decision Analysis - Potential Problem / Opportunity Analysis - Creative Thinking - Problem Solving process - Implementation of the solution.

Unit 4: Business Email Etiquette and Chat 6

Emails Etiquette: Share format/ signature - Emails etiquette - dos and don’ts.

Unit 5: Basics of Finance 6

Accounting systems and how transactions are recorded - Financial statements: Profit & Loss account - balance sheet - cash flow statement - Fixed assets - depreciation and the capitalization of software development expense - Working capital and cash management - Using ratio analysis to assess corporate health and performance - Funding the business: equity - debt and other aspects - Budgeting & Forecasting – capex – apex - Designing a flexible budget - Capital expenditure appraisal and approval

Embedded Programming

Experiment No	Topic	Experiment Details	Type
Experiment 1	Embedded Prog.	Implement C program showcasing Filehandling I/O	Individual
Experiment 2	Embedded Prog.	Implement structures and nested structures	Individual
Experiment 3	Embedded Prog.	Implement Pointers	Individual
Experiment 4	Embedded Prog.	Implement linked list	Individual
Experiment 5	Embedded Prog.	Implement stack and queue	Individual

IoT middleware and Data Engineering

Experiment No	Topic	Experiment Details	Type
Experiment 1	IoT Middleware	Implement real time IoT data from devices to Azure IoT hub and generate dashboard for real time asset utilization	Individual
Experiment 2	IoT Middleware	create device management module with Java/.NET using Azure IOT hub and store the data in Azure cosmosdb. Deploy Java/.NET in Azure App service/Azure AKS	Individual
Experiment 3	IoT Middleware	Create CI/CD pipeline for Java/.NET using Azure DevOps and deploy to Azure AppService/AKS	Individual
Experiment 4	IoT Middleware	Implement real time IoT data from devices to AWS IoT Core and generate dashboard for real time asset utilization	Individual
Experiment 5	IoT Middleware	Create device management module with Java/.NET using AWS IoT Core and store the data in AWS Timeseries. Deploy Java/.NET in AWS EKS	Individual
Experiment 6	IoT Middleware	Create CI/CD pipeline for Java/.NET using AWS DevOps tools and deploy to AWS EKS	Individual
Experiment 7	Data engineering	Implement real time IoT data analytics with Azure synapse analytics and Pyspark scripts for data transformation	Individual
Experiment 8	Data engineering	Implement batch analytics with Azure synapse analytics and SQL transformation	Individual

Experiment 9	Data engineering	Create Power BI dashboards for Batch analytics	Individual
Experiment 10	Data engineering	Implement real time IoT data analytics with AWS Kinesis streams and Glue for data transformation	Individual
Experiment 11	Data engineering	Implement batch analytics with AWS kinesis streams and Glue for data transformation	Individual