SEMESTER V
SYLLABUS

MACHINE LEARNING

Course Code	24IT301		
Course Category:	Program Core	**Credits:**	04
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-2
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course provides a comprehensive introduction to Machine Learning concepts, techniques, and applications. It covers the fundamental ingredients of machine learning including tasks, models, features, and data preprocessing methods. The course introduces classification, regression, model evaluation, and performance assessment techniques. Students will explore major machine learning algorithms such as decision trees, linear models, support vector machines, clustering techniques, Bayesian learning, artificial neural networks, ensemble learning, and reinforcement learning. Emphasis is placed on understanding theoretical foundations, practical implementation, and real-world problem solving using machine learning methods.

COURSE OBJECTIVES

The objectives of this course are to:

1. Introduce the fundamental concepts, terminology, and workflow of machine learning systems.
2. Enable students to understand data preprocessing, feature representation, and statistical analysis techniques.
3. Develop the ability to apply classification, regression, clustering, and probabilistic learning algorithms to real-world problems.
4. Provide knowledge of advanced machine learning approaches including neural networks, ensemble methods, and reinforcement learning.
5. Equip students with skills to evaluate, compare, and select appropriate machine learning models for different applications.

COURSE OUTCOMES

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

CO1: Summarize the fundamental concepts of machine learning, data representation, statistical analysis, and EDA techniques. *(K2)*

CO2: Apply classification, regression, and model evaluation techniques to analyze and solve machine learning problems. *(K3)*

CO3: Design and implement machine learning models including decision trees, linear models, and support vector machines for predictive analytics. *(K3)*

CO4: Analyze and apply distance-based learning, clustering methods, and Bayesian approaches for intelligent data analysis. *(K4)*

CO5: Develop and evaluate advanced machine learning solutions using neural networks, ensemble learning, and reinforcement learning techniques. *(K5)*

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Introduction (9 hrs)

The Ingredients of Machine Learning: Tasks: the problems that can be solved with Machine Learning, Models: the output of Machine Learning, Features: the workhorses of Machine Learning.

Getting to Know Your Data: Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

UNIT II: **Classification and Model Evaluation (9 hrs)**

Binary Classification and Related Tasks: Classification, Assessing classification performance, Visualizing classification performance

Model Evaluation and Selection: Metrics for Evaluating Classifier Performance, Holdout method and Random Subsampling, Cross-Validation, Bootstrap, Model Selection Using Statistical Tests of Significance, Comparing Classifiers Based on Cost–Benefit and ROC Curves.

Beyond Binary Classification: Handling more than two classes, Multi-class classification, Regression

UNIT III: Decision Tree and Linear Models (9 hrs)

Decision Tree Learning: Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree learning algorithm, Inductive bias in decision tree, Issues in decision tree learning.

Linear Models: The least-squares method, Multivariate linear regression, Regularized regression, Using least-squares regression for classification, The perceptron, Support vector machines, Soft margin SVM, Going beyond linearity with kernel methods.

UNIT IV: Distance Based Models and Bayesian Learning (10 hrs)

Distance Based Models: Neighbours and exemplars, Nearest-Neighbour classification, Distance-based clustering- K-means algorithm, clustering around medoids, Hierarchical clustering, From Kernels to Distances.

Bayesian Learning: Introduction, Bayes Theorem-An Example, Bayes Optimal Classifier, Naive Bayes Classifier-An Illustrative Example, Bayesian Belief Networks- Conditional Independence, Representation, Inference, Learning Bayesian Belief Networks.

UNIT V: ANN and Reinforcement Learning (10 hrs)

Artificial Neural Networks: Introduction-Biological Motivation, Neural Network Representations, Appropriate Problems for Neural Network Learning, Multilayer Networks and the Backpropagation Algorithm, Remarks on the Backpropagation Algorithm

Model Ensembles: Bagging and random forests, Boosting-Boosted rule learning, Mapping the ensemble landscape-Bias, variance and margins, other ensemble methods, Meta-learning.

Reinforcement Learning: Introduction, Learning task, Q learning-Q function, an algorithm for learning Q

LIST OF LAB TASKS:

Week-1: Study of Machine Learning Life Cycle and Dataset Sources

Tasks:

- Study problem definition, data collection, preprocessing, model training, evaluation, and deployment.
- Download datasets from: Kaggle , UCI Machine Learning Repository , OpenML

Week-2: Data Preprocessing

Tasks:

- Preprocess the given dataset: Handle missing values. ,Encode categorical variables, Feature scaling (Normalization and Standardization), Detect and remove duplicates.

Week-3: Exploratory Data Analysis (EDA)

Tasks: For a given dataset

- Generate summary statistics.
- Plot histograms, boxplots, pairplots, and correlation heatmaps.
- Identify outliers and feature relationships.

Week-4: Simple and Multiple Linear Regression

Tasks:

- Train a linear regression model.
- Evaluate using MAE, MSE, RMSE, and R^2.

Week-5: Logistic Regression for Classification

Tasks:

- Train logistic regression model.
- Evaluate using confusion matrix, precision, recall, and F1-score.

Week-6: Decision Tree Classifier

Tasks:

- Train decision tree classifier.
- Visualize the tree.
- Analyze overfitting using depth control.

Week-7: Support Vector Machine (SVM)

Tasks:

- Train Linear, Polynomial, and RBF kernel SVMs.
- Compare performance metrics.

Week-8: K-Nearest Neighbors (KNN) and Naïve Bayes Classification

Tasks:

- Train KNN with various values of K.
- Plot accuracy versus K.
- Train Gaussian Naïve Bayes model.
- Evaluate classification performance.

Week-9: K-Means Clustering

Tasks:

- Apply K-Means clustering.
- Use Elbow Method to determine optimal K.
- Visualize clusters.
- Generate dendrogram with Hierarchical Clustering
- Compare with K-Means results.

Week-10: Artificial Neural Network (ANN)

Tasks:

- Design ANN using TensorFlow Keras.
- Train and evaluate the model.
- Plot accuracy and loss curves.

Week-11 &12: Capstone Project

Tasks: Develop and Evaluate the Models by considering a Problem Statement

TEXT BOOK(S):

[1]. Machine Learning: The art and Science of algorithms that make sense of data, Peter Flach, Cambridge University Press, 2012

[2]. Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Education

[3]. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.

REFERENCE BOOKS:

[1]. M N Murthy, V S Ananthanarayana, Machine Learning Theory and Practice, Universities Press (India), 2024
[2]. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor Flow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition
[3]. Stephen Marsland, Machine Learning – An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014
[4]. Ethem Alpaydın, Introduction to machine learning, second edition, MIT press.

E-RESOURCES AND OTHER DIGITAL MATERIAL

[1]. Kevin Murphy,"Machine Learning:A Probabilistic Perspective" , MIT Press, 2012, https://www.cs.ubc.ca/~murphyk/MLbook/pml-intro-5nov11.pdf
[2]. Machine Learning by Andrew Ng, Stanford University https://www.coursera.org/learn/machine-learning
[3]. Professor Prof. Sudeshna Sarkar, IIT Kharagpur "Introduction to machine learning" https://nptel.ac.in/courses/106105152
[4]. Professor Carl Gustaf Jansson, KTH, Video Course on Machine Learning https://nptel.ac.in/noc/individual_course.php?id=noc19-cs35

CLOUD COMPUTING

Course Code	24IT302		
Course Category:	Program Core	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	Computer Networks	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course provides a comprehensive understanding of cloud computing concepts, architectures, and services. It covers cloud deployment models, virtualization, distributed systems, cloud storage, and security aspects. Advanced topics such as serverless computing, IoT integration, and edge computing are also introduced to prepare students to develop modern cloud-based applications.

COURSE OBJECTIVES

1. Introduce the fundamental concepts, characteristics, and evolution of cloud computing along with deployment and service models.
2. Understand the principles of Parallel and Distributed computing that form the foundation of cloud systems.
3. Explain Virtualization techniques and the role of Cloud storage providers such as Amazon Web Services, Google Cloud, and Microsoft Azure in cloud application development.
4. Explore the cloud platforms in industry and examine real-world applications.
5. Introduce the cloud security mechanisms and advanced topics in modern cloud computing environments.

COURSE OUTCOMES

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

CO1: Illustrate the fundamentals of cloud computing, services and deployment models of modern Computing environments [K2]

CO2: Analyze parallel and distributed computing concepts applicable to cloud systems. [K4]

CO3: Apply virtualization techniques and cloud storage services to build and deploy scalable cloud applications. [K3]

CO4: Analyze major cloud platforms applicable for real-world problems. [K4]

CO5: Explore cloud security mechanisms and advanced cloud computing technologies for securing cloud-based applications and services [K4]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES

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

COURSE CONTENT

UNIT I: Introduction (7 hrs)
Overview and definition of Cloud Computing, Cloud Computing reference model, Types of Services (IaaS, PaaS, SaaS), Database as a Service, Storage as a Service, Cloud Deployment models (public, private, hybrid), Cloud Computing characteristics and benefits, Cloud Service Providers, Data center Design.

UNIT II: **Principles of Parallel and Distributed Computing (8 hrs)**
Parallel Computing: Elements of Parallel Computing, Hardware Architecture for Parallel Processing, Approaches to Parallel Programming, Levels of Parallelism.
Distributed Computing: Elements of Distributed Computing, Components of a Distributed System, Architectural Styles for Distributed Computing, Models for Inter-Process Communication, Technologies for Distributed Computing.

UNIT III: Virtualization & Cloud Storage (9 hrs)
Virtualization: Virtualization in Cloud Computing, Pros and Cons of Virtualization, Xen: Para Vrtualization, VMWare: Full Virtualization, Microsoft Hyper-V.
Cloud Storage: Overview, Cloud Storage Providers – Amazon S3, Google Bigtable Data store, Google Drive, Apple iCloud.
Software plus services: Overview, Mobile Device Integration, Providers-Adobe AIR, Apple iPhone SDK, Microsoft Online.

UNIT IV: Cloud Platforms & Applications(9 hrs)
Cloud Platforms in Industry:
Amazon Web Services, Compute Services, Storage and Communication Services, Google App Engine, Architecture and core concepts, Application Life Cycle, Microsoft Azure, Azure Core Concepts, SQL Azure.
Cloud Applications:
Scientific applications: Healthcare, Geoscience, Business and consumer applications: CRM, Salesforce.com, Productivity: Dropbox and iCloud, Social Networking: Facebook.

UNIT V: Cloud Security (8 hrs)
Cloud Security: Importance of Cloud Security, Cloud Security Architecture, Cloud Shared Responsibility Model, Security in Cloud Deployment Models.
Advanced Concepts in Cloud Computing:
Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (AWS Lambda) and open-source (OpenFaaS), Integration of Cloud with IoT.

Text Book(s):
[1]. Mastering Cloud Computing, 2nd edition, Rajkumar Buyya, Christian Vecchiola,Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, Mc Graw Hill, 2024.
[2]. Velte T. Antony, Velte J. Toby., Elsenpeter Robert, “Cloud Computing: A Practical Approach”, Tata McGraw- Hill, 2010.

Reference Books:

[1]. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
[2]. Essentials of cloud Computing, K. Chandrasekhran, CRC press, 2014.

E-resources and other digital material

[1]. Prof. Soumya Kanti Ghosh, IIT Kharagpur, "Cloud Computing", 2025, https://nptel.ac.in/courses/106105167.
[2]. Udemy – "Cloud Computing: The Complete Guide for Beginners", 2025, https://www.udemy.com/course/intro-to-cloud-computing/.
[3]. AWS Training & Certification – Cloud Essentials, Serverless & Security, 2024, https://aws.amazon.com/training.
[4]. Microsoft Learn – Azure Cloud Fundamentals, 2024, https://learn.microsoft.com/en-us/training.
[5]. Google Cloud Skills Boost – Cloud Engineering & Serverless Development, 2024, https://www.cloudskillsboost.google.

ADVANCED JAVA PROGRAMMING

Course Code	**24IT303**		
Course Category:	Program Core	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	Java Programming	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course provides a comprehensive understanding of enterprise-level Java web application development using JDBC, Servlets, and Spring Boot. It covers database connectivity with JDBC, client-server communication with Java Servlets, and modern application development with the Spring Boot framework. The course also covers Spring Data JPA for relational database operations and application deployment techniques. Students gain practical knowledge of developing scalable, secure, and maintainable web applications using industry-standard Java technologies and frameworks.

COURSE OBJECTIVES

1. Understand the fundamentals of JDBC and perform database connectivity operations in Java applications.
2. Develop dynamic web applications using Java Servlets and manage client-server interactions effectively.
3. Understand the architecture, features, and development workflow of Spring Boot applications.
4. Build enterprise applications using Spring Boot annotations, Maven, and dependency management tools.
5. Implement database operations using Spring Data JPA and deploy web applications on application servers.

COURSE OUTCOMES (COs)

CO1: Summarize the concepts of JDBC, JDBC drivers, database connectivity and statement handling mechanisms. (K2)

CO2: Develop Java Servlet-based web applications for handling client requests, cookies, and session tracking. (K3)

CO3: Illustrate the architecture, features, and setup process of Spring Boot applications. (K2)

CO4: Build Spring Boot applications using annotations, Maven, and application configuration techniques. (K3)

CO5: Implement relational database operations using Spring Data JPA and deploy web applications on servers. (K3)

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: JDBC (7 hrs)

The concept of JDBC, JDBC Driver Types, JDBC Packages, A Brief Overview Of The JDBC Process, Database Connection, Associating The JDBC/ODBC bridge with the Database, Statement objects-Statement, Prepared Statement and Callable Statement, Result Set.

UNIT II: Java Servlets (9 hrs)

Java Servlets and common gateway interface programming, benefits of using a java servlets, simple java servlet, anatomy of a java servlet, deployment descriptor, reading data from a client, sending data to a client, working the cookies, tracking sessions.

UNIT III: Getting Started with Spring Boot (8 hrs)

Structure, objectives, introduction, features, advantages of Spring Boot, Breaking the monolithic way of developing software, system requirements, setting up of the environment, the 12-factor app, Spring initializer.

UNIT IV: Developing Spring Boot Application (9 hrs)

Starting with Spring initializer, Build tools, building an application using Maven, understanding the entry point class and Spring Boot Application annotation.

Spring Boot Annotations (8 hrs)

Java Annotations, existence of spring annotations, Spring and Spring Boot annotations, Stereotype annotations, Spring Boot Annotations.

UNIT V: Working with Spring Data JPA (8 hrs)

Accessing relational data using JDBC Template and Spring data JPA in memory database, Spring data JPA with MySQL, Query methods in Spring data JPA.

Text Books

1. James Keogh, *"J2EE: The Complete Reference"*, 1st Edition, McGraw Hill Education, 2002.
2. Shagun Bakliwal, *Hands-on Application Development using Spring Boot*, BPB Publications, First Edition, 2022.

Reference Books

1. Craig Walls, *Spring in Action*, Sixth Edition, MEAP Edition, Manning Early Access Program, Version 4, 2021.
2. Mark Heckler, *Spring Boot: Up and Running*, O'Reilly Media, 2021.

E-Resources and Other Digital Material

1. Ranga Karanam, *Java Servlets and JSP - Build Java EE (JEE) app in 25 Steps*, Available:
   Udemy Course
2. Spring Official Documentation, Available:
   Spring Boot Official Documentation
3. *Advanced Java Programming* by Infinite Skills, Available:
   Udemy Advanced Java Programming
4. Derek Parsons, *Spring MVC, Spring Boot and Rest Controllers*, LearnQuest, Available:
   Coursera Course
5. Ranga Karanam, *Spring Framework Master Class - Java Spring the Modern Way*, Available:
   Udemy Spring Framework Master Class

ARTIFICIAL INTELLIGENCE TECHNIQUES

Course Code	**24IT360A**		
Course Category:	Interdisciplinary Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	Mathematics, Probability and statistic, Programming and Problem Solving	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course introduces the fundamental concepts of Artificial Intelligence (AI) and its applications in engineering. It covers intelligent agents, problem-solving techniques, knowledge representation, machine learning, and modern AI applications such as natural language processing and computer vision. The course emphasizes both theoretical foundations and practical implementation of AI systems..

COURSE OBJECTIVES

1. Understand core concepts of Artificial Intelligence and intelligent systems
2. Apply search and reasoning techniques to solve engineering problems
3. Learn foundational machine learning algorithms and models
4. Explore AI applications in real-world engineering domains
5. Analyze ethical, social, and industrial implications of AI

COURSE OUTCOMES

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

CO1: Explain AI fundamentals and intelligent agent models. [K2]

CO2: Apply search algorithms and problem-solving strategies. [K3]

CO3: Analyze knowledge representation and reasoning methods. [K3]

CO4: Implement basic generative AI models and prompt injection. [K4]

CO5: Evaluate AI applications and ethical considerations. [K4]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Introduction to AI (7 hrs)

Definition and History of AI, Applications and Impact of AI in Various Domains,

Types of AI: Narrow AI, General AI, and Super AI, Intelligent Agents, Agents and Environments,
AI in Problem Solving: State Space Representation, AI Ethics and Challenges

UNIT II: Problem-solving (8 hrs)
Problem-Solving Agents, Search Algorithms, Un informed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Search in Complex Environments, Local Search and Optimization Problems

UNIT III: Knowledge and Reasoning (9 hrs)
Propositional Logic: Propositional Theorem Proving, Effective Propositional Model Checking, Agents Based on Propositional Logic,
First-Order Logic: Representation Revisited, Syntax and Semantics of First Order Logic, Knowledge Engineering in First-Order Logic.

UNIT IV: Introduction of Generative AI (9 hrs)
Introduction to Generative AI: Evolution of Generative Models, Foundations of Large Language Models (LLMs), Applications of Generative AI.
Prompt Engineering: Prompt Engineering Concepts and Techniques, Types of Prompts: Zero-shot, One-shot and Few-shot Prompting, Prompt Injection Attacks and Security Risks

UNIT V: Planning and Emerging AI Topics (8 hrs)
Automated and Classical Planning, Hierarchical Planning, Planning under Uncertainty, Analysis of Planning Approaches, Explainable AI (XAI), Federated Learning, Edge AI.

Text Book(s):

[1] Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN: 10:0136042597
[2] Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education (India), 2013, ISBN: 978-1-25-902998-1
[3] Elaine Rich, Kevin Knight and Nair, "Artificial Intelligence", TMH, ISBN-978-0-07-0087705.
[4] James Phoenix and Mike Taylor, "Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs", O'Reilly Media, 2024, ISBN: 978-1-098-15343-4.

Reference Books:

[1] Nilsson Nils J, "Artificial Intelligence: A new Synthesis", Morgan Kaufmann Publishers Inc. San Francisco, CA, ISBN: 978-1-55-860467-4
[2] Patrick Henry Winston, "Artificial Intelligence", Addison-Wesley Publishing Company, ISBN: 0 201-53377-4
[3] Andries P. Engelbrecht-Computational Intelligence: An Introduction, 2nd Edition-Wiley India ISBN: 978-0-470-51250-0
[4] Dr. Lavika Goel, "Artificial Intelligence: Concepts and Applications", Wiley publication, ISBN:9788126519934

E-resources and other digital material

[1] **Prof. Deepak Khemani,** Department of Computer Science & Engineering, IIT Madras, NPTEL, Lecture Series on Artificial Intelligence, 2014.
https://nptel.ac.in/courses/106106126

[2] **Sander Schulhoff et al.,** ***Learn Prompting — The Prompt Engineering Guide***
https://learnprompting.org/docs/introduction

[3] **Prof. Viraj Kumar (IISc Bangalore), NPTEL / SWAYAM,** ***Programming with Generative AI*****, 2025.**
https://onlinecourses.nptel.ac.in/noc25_cs137/preview

[4] **Stuart Russell and Peter Norvig,** Artificial Intelligence: A Modern Approach, 4th Edition – companion slides, exercises, and supplementary material. Official AIMA website.
http://aima.cs.berkeley.edu

DATA STRUCTURES

Course Code	24IT360B		
Course Category:	Interdisciplinary Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	Programming using C	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course covers fundamental concepts of data structures and algorithm design for efficient problem solving. It includes linear structures such as arrays, stacks, and queues, non-linear structures like binary trees and binary search trees, sorting techniques, and algorithm design methods such as divide and conquer, greedy approach, and dynamic programming. The objective of this course is to enable students to develop efficient and optimized solutions for real-world problems using appropriate data structures and algorithms.

COURSE OBJECTIVES

1. Understand fundamental concepts of data structures, algorithms, and their role in problem solving, including time and space complexity.
2. Analyze and apply basic searching and sorting techniques for efficient data processing
3. Apply linear data structures such as stacks, queues, and linked lists along with their operations and applications.
4. Understand and implement non-linear data structures like trees, including binary search trees and AVL trees.
5. Understand graph concepts and apply traversal techniques and greedy methods for solving optimization problems.

COURSE OUTCOMES

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

CO1: Illuminate fundamental concepts of data structures, algorithm design, and analyze time and space complexity of programs. [K2]

CO2: Implement and compare searching and sorting algorithms such as linear search, binary search, bubble sort, merge sort, and quick sort. [K3]

CO3: Design and apply linear data structures including stacks, queues, and linked lists to solve real-world problems like expression evaluation. [K3]

CO4: Construct and perform operations on binary trees, binary search trees, and AVL trees, including insertion, deletion, and traversal. [K3]

CO5: Apply graph traversal and optimization techniques such as DFS, BFS, minimum spanning tree, and shortest path algorithms using greedy methods. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I : Searching and Sorting (8 hrs)

Introduction: Basic Terminology, Classification of Data Structures, Operations on Data Structures, Abstract Data Type, Algorithms, Control Structures Used in Algorithms, Time and Space Complexity.

Searching: Linear Search and Binary Search Techniques

Sorting: Bubble sort, Insertion sort, Selection sort, Merge sort and Quick Sort.

UNIT II : Stacks and Queues (9 hrs)

Stacks: Introduction to Stacks, Array Representation of Stacks, Operations on a Stack, Linked Representation of Stacks, Operations on a Linked Stack, Applications of Stacks- Evaluation of Arithmetic Expressions, Recursion.

Queues: Introduction to Queues, Array Representation of Queues, Linked Representation of Queue, Operations of Queue, Circular Queues, Applications of Queues

UNIT III: Linked Lists (8 hrs)

Linked list: Introduction, Singly linked lists-Traversing, Searching, Inserting, Deleting a Node from a linked list

Doubly linked list: Inserting and deleting a Node from a Doubly linked list.

Circular Linked List: Inserting and deleting a Node from a Doubly linked list.

Applications of Linked Lists: Polynomial representation

UNIT IV: Trees (9 hrs)

Trees–Introduction, Binary trees, Creating a Binary Tree from a General Tree, Traversing a Binary Tree

Efficient Binary Trees – Binary search tree, Operations on Binary Search Trees- Searching, Inserting and Deleting a node.

Balanced search trees-AVL Tree: Definition, Height of an AVL Tree, Representation of an AVL Tree, Searching an AVL Search Tree, Inserting into an AVL Search Tree.

UNIT V: Graphs (7 hrs)

Graphs: The Graph Abstract Data Type-Introduction, Definition, Graph representation (Adjacency Matrix/List), Elementary Graph Operations- Depth First Search (DFS) and Breadth First Search (BFS),

Greedy Algorithm: General Method, Minimum cost spanning Trees, Single-Source Shortest Paths.

Text Book(s):

[1].Horowitz, Sahni, and Anderson-Freed, Fundamentals of Data Structures in C, 2nd edition, Universities Press, 2011.

[2].Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd Edition, Pearson Education,2002

[3]. Reema Thereja, Data Structures using C, Oxford Publications, 2nd Edition, 2014.

.**Reference Books:**

[1]. Thomas H Corman, E Leiserson, Ron Rivest, "Introduction to Algorithms" , MIT Press, 2nd Edition, Jan 2001.

[2]. Alfred V Aho, J D Ullman, J E Hopcroft, "Data Structures and Algorithms" , Addison Wesley Longman, 1983.

[3]. Mark Allen Weiss, "Data Structures in C++", Addison Wesley Longman, 2nd Edition, 1998.

[4]. Horowitz E and Sahni S, "Fundamentals of Computer Algorithms", Computer Science Press, 1984.

E-resources and other digital material

[1].SudarshanIyengar: IIT Ropar (12, August, 2018). Data Structures and Algorithms[NPTEL]. Available: http://nptel.ac.in/

[2].Erik Demaine, (12, May, 2018). Advanced Data Structures [MIT-OpenCourseWare]. Available: http://ocw.mit.edu/

[3].HackerRank – Practice problems on data structures. Available: https://www.hackerrank.com/domains/data-structures

[4]. LeetCode – Advanced problem-solving (https://leetcode.com/problemset/all/

[5].https://www.coursera.org/specializations/data-structures-algorithms

WEB TECHNOLOGIES AND FRONT END DEVELOPMENT

Course Code	**24IT360C**		
Course Category:	Interdisciplinary Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course introduces the fundamentals of web technologies and front-end development using HTML5, CSS3, and JavaScript. Students will learn to design responsive, interactive, and user-friendly web pages with modern styling and dynamic functionality. The course covers data handling with JSON & AJAX, along with deployment on platforms like GitHub Pages and Netlify. It culminates in a mini-project to build practical web applications relevant for engineering domains.

COURSE OBJECTIVES

1. Understand the fundamentals of web technologies, Internet architecture, and client-side scripting.
2. Apply HTML5 and CSS3 techniques to design structured and responsive web pages.
3. Develop interactive web applications using JavaScript, DOM manipulation, and event handling.
4. Apply JSON and AJAX/Fetch concepts for asynchronous data handling and dynamic web interaction.
5. Design and deploy simple real-world web applications using modern tools and hosting platforms.

COURSE OUTCOMES

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

CO1: Summarize the fundamentals of web technologies, Internet architecture, and HTML5 concepts for developing structured web pages. [K2]

CO2: Apply HTML5 and CSS3 techniques to create structured, styled, and responsive web pages. [K3]

CO3: Develop interactive web pages using JavaScript programming constructs, functions, arrays, objects, and DOM manipulation techniques. [K3]

CO4: Implement event handling, JSON, and AJAX/Fetch concepts for dynamic data handling and enhanced user interaction in web applications. [K4]

CO5: Design and deploy simple real-world responsive web applications using modern development tools and hosting platforms. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Web Foundations and HTML5 Basics (8 hrs)

Internet & Web Basics: Introduction, The Internet in Industry and Research, HTML5, CSS3, JavaScript, Evolution of the Internet and World Wide Web , Web Basics , Multitier application architecture. client-side vs server-side scripting.

HTML5 Fundamentals: Introduction, Editing HTML5, First HTML5 Example, W3C HTML5 Validation Service, Headings, Linking, Images, Special Characters and Horizontal Rules, Lists, Tables, Forms, Internal linking.

UNIT II: Advanced HTML5 and CSS3 Styling (8 hrs)

HTML5 Advanced Features: New HTML5 Form input Types, input and data list Elements and autocomplete Attribute, Page-Structure Elements.

CSS Fundamentals: Inline Styles, Embedded Style Sheets, Conflicting Styles, Linking External Style Sheets, Positioning Elements: Relative Positioning, span, Box Model and Text Flow, Media Types and Media Queries, Drop-Down Menus.

UNIT III: CSS3 Effects and JavaScript Fundamentals (9 hrs)

CSS3 Enhancements: Introduction, Text Shadows , Rounded Corners , Color, Box Shadows. Image Borders, Animation; Selectors, Transitions and Transformations.

JavaScript Basics: First Script, Modifying First Script, Obtaining User Input with prompt Dialogs, Memory Concepts , Arithmetic, Decision Making: Equality and Relational Operators.

Control Statements: Control Statements, if Selection Statement, if…else Selection Statement, while Repetition Statement, Counter-Controlled Repetition, Assignment Operators, Increment

and Decrement Operators, switch Multiple-Selection Statement, do…while Repetition Statement, break and continue Statements , Logical Operators .

UNIT IV: JavaScript Programming and Dynamic Interaction (9 hrs)

Functions: Function Definitions. Programmer-Defined Functions, Random Number Generation, Scope Rules, JavaScript Global Functions, Recursion, Recursion vs. Iteration.

Arrays and Objects: Arrays, Declaring and Allocating Arrays, Examples Using Arrays, introduction to JavaScript objects and Math object, String object, Date object, document Object.

Document Object Model (DOM): Modeling a Document: DOM Nodes and Trees, Traversing and Modifying a DOM Tree, DOM Collections, Dynamic Styles.

Event Handling: Reviewing the load Event, Event mousemove and the event Object, Rollovers with mouseover and mouseout, Form Processing with focus and blur, Event Bubbling.

UNIT V: Data Handling, AJAX, and Web Deployment (8 hrs)

Ajax-Enabled Rich Internet Applications with XML and JSON: Introduction, Rich Internet Applications (RIAs) with Ajax, History of Ajax, "Raw" Ajax Example Using the XMLHttpRequest Object, Creating a Full-Scale Ajax-Enabled Application.

Web Deployment: Domain names, hosting concepts, and deploying websites using platforms like GitHub Pages or Netlify.

Mini Project: Development of a simple real-world web application (portfolio or domain-based).

Text Book(s):

[1]. Paul Deitel, Harvey Deitel, Abbey Deitel, Internet & World Wide Web: How to Program, 5th Edition, Pearson, 2020.

Reference Books:

[1] Jon Duckett, HTML and CSS: Design and Build Websites, Wiley, Reprint 2023.

[2] Jon Duckett, JavaScript and JQuery: Interactive Front-End Web Development, Wiley, Reprint 2022.

[3] Uttam K. Roy, Web Technologies, 2nd Edition, Oxford University Press, 2023.

E-resources:

[1]. Dr. Indranil Sengupta, Professor, IIT Kharagpur, "Internet Technologies", 2024. Link: NPTEL Internet Technologies

[2]. Yaakov Chaikin, Adjunct Professor, Johns Hopkins University, "HTML, CSS, and Javascript for Web Developers", 2023.

[3]. Haris Khan, Software Educator, "Complete Web Development Course", 2024. Link: CodeWithHarry YouTube Channel

DATA VISUALIZATION

Course Code	24IT360D		
Course Category:	Interdisciplinary Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course focuses on the principles and techniques of data visualization and interpretation for analyzing complex datasets. It provides knowledge of visualization design, chart selection, and effective data presentation for meaningful insights. The course also offers practical exposure to Tableau for creating interactive dashboards and analytical reports. The objective of this course is to enable students to design effective visualization solutions for research, business intelligence, and real-world data analytics applications.

COURSE OBJECTIVES

1. Outline the fundamental concepts, principles, and methodologies of data visualization and interpretation for analyzing complex datasets.
2. Analyze different data visualization techniques and chart types suitable for diverse research and real-world applications.
3. Apply visualization design principles and data representation methods to create meaningful and effective visualizations.
4. Develop interactive dashboards and analytical reports using visualization tools such as Tableau.
5. Evaluate visualization solutions for effective communication, interpretation, and decision-making in business intelligence and data analytics applications.

COURSE OUTCOMES

CO1: Summarize the fundamentals, principles, and methodologies of data visualization and interpretation. [K2]

CO2: Apply visualization design techniques and chart selection methods to represent complex datasets effectively. [K3]

CO3: Analyze different data visualization methods and choose appropriate visualization solutions for research and real-world applications. [K4]

CO4: Develop interactive dashboards and analytical reports using visualization tools such as Tableau. [K3]

CO5: Design real-time interactive visualization systems for effective communication and decision-making in business intelligence and data analytics applications. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Basics of Data Visualization (9 hrs)

The Context of Data Visualization: Visualization as a Discovery Tool, The Bedrock of Visualization Knowledge, Defining Data Visualization, Visualization Skills for The Masses, The Data Visualization Methodology.

Setting the Purpose and Identifying Key Factors: Establishing Intent – The Visualization's Function, Establishing Intent – The Visualization's Tone, Key Factors Surrounding a Visualization Project, The " Eight Hats" Of Data Visualization Design.

UNIT II: Design Options (8 hrs)

Conceiving and Reasoning Visualization Design Options: Data visualization design is all about choices, The visualization anatomy – data representation, The visualization anatomy –data presentation. Using visual analysis to find stories , An example of finding and telling stories

UNIT III: Data Visualization Methods (8 hrs)

Taxonomy of Data Visualization Methods: Data visualization methods, Choosing the appropriate chart type, Assessing hierarchies and part-to-whole relationships.

Constructing and Evaluating Your Design Solution: For constructing visualizations, technology matters, the construction process, Approaching the finishing linePost-launch evaluation.

Case Studies on real-time applications.

UNIT IV: Introduction to Connecting to Data and Aggregation (9 hrs)

An Introduction to Connecting to Data: An Introduction to Connecting to Data in Tableau, Shaping Data for Use with Tableau, Getting a Lay of the Land: Tableau
Terminology, View the Underlying Data, View the Number of Records,

Five Ways to Make a Bar Chart/An Introduction to Aggregation: Five Ways to Create a Bar Chart in Tableau.An Introduction to Aggregation in Tableau, Line Graphs, Independent Axes, And Date Hierarchies, How to Make a Line Graph in Tableau, Independent Axes in Tableau, Date Hierarchies in Tableau, Marks Cards, Encoding, and Level of Detail, An Explanation of Level of Detail, An Introduction to Encoding, Label and Tooltip Marks Cards.

UNIT V: Encoding and Filters (8 hrs)

Marks Cards, Encoding, and Level of Detail : An Explanation of Level of Detail, An Introduction to Encoding ,Label and Tooltip Marks Cards

An Introduction to Filters : Dimension Filters in Tableau ,Measure Filters in Tableau ,More Options with Filters , Macro Filters

Case Studies on real-time applications.

Text Book(s):

[1]. Andy Kirk, "Data Visualization: a successful design process", Packt Publishing (26 December 2019)

[2]. Ryan Sleeper, Practical Tableau, O'Reilly Media, Inc. April 2018

Reference Books:

[1]. Ian Sommerville , Software Engineering, ,10th Edition, Pearson.

[2]. Deepak Jain , Software Engineering, Principles and Practices, Oxford University Press.

[3]. Booch G., Rumbaugh J. and JacobsonsI , The Unified Modelling Language User Guide, 2nd Edition, Addison Wesley, 2005

E-resources and other digital material

[1]. Dr. Gauravdixit, Department of Management Studies, Indian Institute of Technology, Roorkee: https://Nptel.Ac.In/Courses/110107092/7,2017

[2]. P Adam Marcus, and Eugene Wu. RES.6-009 How to Process, Analyze and Visualize Data. January IAP 2012. Massachusetts Institute of Technology: MIT Open Courseware, https://ocw.mit.edu.,2012.

[3]. Prof.Shankarnarasimhan,Ragunathan, Rengasamy,Iit Madras Data Visualization In R Basic Graphics, https://Nptel.Ac.In/Courses/106106179/11,2016

[4]. Statistics and Visualization for Data Analysis and Inference, Dr. Ed Vul, Dr. Mike Frank, Massachusetts Institute of Technology,https://ocw.mit.edu/courses/res-9-0002-statistics-and-visualization-for-data-analysis-and-inference-january-iap-2009/resources/lec1_visulzatn/

DATA WAREHOUSING AND MINING

Course Code	24IT310A		
Course Category:	Professional Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	Database Management Systems	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course introduces the fundamentals of data warehousing and data mining, covering data warehouse design, OLAP operations, and implementation for decision support. It explores core data mining concepts, technologies, applications, and challenges, along with techniques for understanding data through statistical analysis, visualization, and similarity measures. Emphasis is placed on data preprocessing methods such as cleaning, integration, transformation, and discretization. The course also covers key data mining techniques including association rule mining, classification, and clustering using algorithms like Apriori, FP-Growth, decision trees, Naïve Bayes, K-means, hierarchical clustering, and DBSCAN, enabling students to extract meaningful patterns and insights from large datasets.

COURSE OBJECTIVES

1. Understand data warehousing concepts and OLAP operations.
2. Learn fundamentals and applications of data mining.
3. Analyze data using statistical and visualization techniques.
4. Apply data preprocessing methods.
5. Implement association rule mining algorithms.
6. Apply classification techniques for prediction.
7. Explore clustering methods for pattern discovery.

COURSE OUTCOMES

At the end of the course, the student will be able to
CO1: Outline the basic concepts of Warehousing and Mining. [K2]
CO2: Derive interesting patterns and associations in datasets. [K3]
CO3: Design classifier models to predict future trends. [K4]
CO4: Apply unsupervised learning techniques for a given application. [K3]
CO5: Apply data preprocessing techniques such as data cleaning, data integration, data transformation, data reduction effective data mining and analysis. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Data Warehousing Introduction (7 hrs)

Data Warehousing and Online Analytical Processing: Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation.

UNIT II: **Data Mining Introduction (8 hrs)**

Data Mining Introduction: Data Mining, kinds of data, Kinds of Patterns, Technologies, Applications, Major issues.

Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

UNIT III: Association Analysis (8 hrs)

Association Analysis-Basic Concepts and Algorithms: Problem Definition, Frequent Itemset Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm, Evaluation of Association Patterns.

Association Analysis: Advanced Concepts: Handling Categorical Attributes, Handling Continuous Attributes.

UNIT IV: Decision Tree and Rule-Based Classification (9 hrs)

Classification: Basic Concepts, General Approach to solving a classification problem.

Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction**.**

Rule-Based Classification: Using IF-THEN Rules for Classification, Rule Extraction from a Decision Tree.

UNIT V: Cluster Analysis and Outlier Detection (9 hrs)

Cluster Analysis: What Is Cluster Analysis, Requirements for Cluster Analysis, Overview of Basic Clustering Methods, Partitioning Methods: k-Means: A Centroid-Based Technique, Evaluation of Clustering, Assessing Clustering Tendency, Determining the Number of Clusters, Measuring Clustering Quality.

Outlier Detection: Outliers and Outlier Analysis, Outlier Detection Methods, Clustering-Based Approaches.

Text Book(s):

[1]. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
[2]. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:

[1]. Data Mining: VikramPudi and P. Radha Krishna, Oxford Publisher.
[2]. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press, 2013.

E-resources and other digital material

[1]. Prof. Pabitra Mitra, NPTEL course http://onlinecourses.nptel.ac.in/noc17_mg24/preview
[2]. Jiawei Han, John C. Hart, ChengXiang Zhai, Department of Computer Science, University of Illinois, https://www.coursera.org/specializations/data-mining

INTRODUCTION TO QUANTUM COMPUTING

Course Code	**24IT310B**		
Course Category:	Professional Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	Linear Algebra, Discrete Mathematics	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course provides a comprehensive foundation in quantum computing, covering quantum mechanical principles, mathematical preliminaries, and quantum circuit models. Students will explore vector spaces, Braket notation, quantum gates, and quantum parallelism. The course progresses through key quantum algorithms Deutsch-Jozsa, Grover's Search, Shor's Algorithm, and the Quantum Fourier Transform and concludes with quantum error correction, cryptographic protocols, and emerging real-world quantum applications.

COURSE OBJECTIVES

1. Understand quantum mechanics fundamentals including vector spaces, Bra-ket notation, postulates, composite systems, and quantum states.
2. Analyze quantum circuit models, quantum gates, Turing machine concepts, and complexity classes in the context of quantum computation.
3. Apply circuit optimization techniques and introductory quantum algorithms including the Deutsch and Deutsch Jozsa algorithms.
4. Implement Grover's search algorithm, Shor's algorithm, and the Quantum Fourier Transform for solving complex computational problems.
5. Evaluate quantum error correction models, fault-tolerant computation, and quantum cryptography protocols for real-world quantum applications.

COURSE OUTCOMES

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

CO1: Explain fundamental concepts of quantum computing including vector spaces, Braket notation, measurements, composite systems, Bloch sphere, and mixed states. [K2]

CO2: Analyze quantum circuit models, quantum gates, Turing machine concepts, and complexity classes related to quantum computation. [K4]

CO3: Apply Deutsch and Deutsch–Jozsa algorithms and evaluate circuit quality measures and optimization techniques for quantum parallelism. [K3]

CO4: Implement Grover's search algorithm, Shor's algorithm, and the Quantum Fourier Transform to solve unstructured search and integer factoring problems. [K3]

CO5: Evaluate quantum error correction models, fault-tolerant computation, quantum cryptography protocols, and variational quantum algorithms for real-world applications. [K5]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES

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

COURSE CONTENT

UNIT I: Introduction (8 hrs)

Foundations of Quantum Computing : Need for Quantum Computing and fundamental concepts, Vector spaces, Complex numbers and mathematical preliminaries, Hilbert Spaces, Braket notation, Postulates of quantum mechanics, Measurements, Composite systems, Bloch sphere, Entanglement, Pure and Mixed states.

UNIT II: Gates and Circuit Design (7 hrs)

Quantum Gates and Circuit Design: Geometry of quantum states, Complexity classes, Turing machine and Turing machine concepts, Quantum gates, Quantum circuits, Quantum circuits design.

UNIT III: Algorithms (8 hrs)

Quantum Algorithms : Quantitative measures of circuit, Analysis of quality of circuits, Circuit optimization, Introduction to quantum parallelism, Deutsch Algorithm, Deutsch Jozsa Algorithm, Simon's Algorithm.

UNIT IV: Search and Transform Algorithms (9 hrs)

Quantum Search and Transform Algorithms : Introduction to Grover's algorithm, Detailed walkthrough of Grover's algorithm, Geometric visualization of Grover's iterations, Grover's search applied to unstructured databases, Quantum Teleportation, Shor's Algorithm, Quantum Fourier Transform.

UNIT V: Quantum Applications (9 hrs)

Quantum Applications and Error Correction: Introduction to quantum applications, Research challenges of quantum computing, Quantum Complexity Theory and Lower Bounds, Eigenvalue Estimation and Variational Quantum Eigensolvers, Quantum Cryptography, Quantum Error Correction Models, Fault Tolerant Quantum Computation.

Text Book(s):

[1]. Phillip Kaye, Raymond Laflamme and Michele Mosca, "An Introduction to Quantum Computing", Oxford University Press, 1st Edition, 2007.

[2]. Michael A. Nielsen and Isaac L. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 10th Anniversary Edition, 2010.

Reference Books:

[1]. N. David Mermin, "Quantum Computer Science: An Introduction", Cambridge University Press, 2007.
[2]. Jack D. Hidary, "Quantum Computing: An Applied Approach", Springer, 2nd Edition, 2021.
[3]. Wolfgang Scherer, "Mathematics of Quantum Computing", Springer, 2019.
[4]. Mikio Nakahara and Tetsuo Ohmi, "Quantum Computing: From Linear Algebra to Physical Realizations", CRC Press, 2008.

E-resources and other digital material

[1]. Shor, P., Department of Mathematics, Massachusetts, Institute of Technology (MIT), USA. MIT OpenCourseWare Quantum Computation (18.435J / 8.370), Fall 2003. https://ocw.mit.edu/courses/18-435j-quantum-computation-fall-2003
[2]. Mandayam, P., Ray, A., and Raghunathan, S., IIT Madras & IBM Research, India., NPTEL Introduction to Quantum Computing: Quantum Algorithms and Qiskit –. https://nptel.ac.in/courses/106106143
[3]. aronson, S., Department of Computer Science, The University of Texas at Austin (UAustin), USA., Introduction to Quantum Information Science – Lecture notes https://www.scottaaronson.com/qclec.pdf

REMOTE SENSING ESSENTIALS

Course Code	24IT310C		
Course Category:	Professional Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course introduces the fundamental principles and techniques of remote sensing for Earth observation and geospatial analysis. It covers the physics of electromagnetic radiation, sensor characteristics, image interpretation, and digital image processing techniques. The course also emphasizes classification methods and practical applications in land use/land cover mapping, agriculture, forestry, and environmental monitoring. The objective is to equip students with the ability to analyze satellite imagery and derive meaningful geospatial information for real-world applications.

COURSE OBJECTIVES

1. Understand the basic principles of remote sensing, including electromagnetic radiation and sensor characteristics.
2. Analyze different types of remote sensing systems and satellite platforms for data acquisition.
3. Apply visual interpretation techniques and digital image processing methods for extracting information from satellite imagery.
4. Develop skills in image classification and accuracy assessment for land use/land cover mapping.
5. Evaluate the applications of remote sensing in agriculture, forestry, and environmental monitoring.

COURSE OUTCOMES

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

CO1: Explain the fundamental concepts of remote sensing, including energy interactions, sensors, and platforms. **[K2]**

CO2: Interpret satellite images using visual interpretation elements and generate thematic maps. **[K3]**

CO3: Apply digital image processing techniques such as preprocessing, enhancement, and transformations. **[K3]**

CO4: Analyze image classification methods and assess classification accuracy. **[K4]**

CO5: Explore remote sensing techniques in real-world applications **[K4]**

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

COs / POs	PO1	PO2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO2
CO1	3	2	1	1								1	
CO2		3			3								2
CO3	2	1	2	2	2							1	
CO4						1						2	
CO5	2		2		1	2							1

COURSE CONTENT

UNIT I: Introduction (9 hrs)

Concept of Remote Sensing: Introduction, Definition of Remote Sensing, Data, Remote Sensing Process, Source of Energy, Interaction with Atmosphere, Interaction with Target, Recording of Energy by Sensor, Transmission-Reception and Processing, Interpretation and Analysis.

Types of Remote Sensing and Sensor Characteristics: Introduction, Types of remote sensing, Characteristics of images, Remote Sensing Satellites, Sensor Resolutions, Orbital Characteristics of Satellite, Concept of Swath, Concept of Nadir.

UNIT II: Image Interpretation (8 hrs)

Visual Image Interpretation: Introduction, Image Interpretation, Elements of Visual Image Interpretation, Generation of Thematic maps.

Thermal Image Interpretation: Radiant versus Kinetic Temperature, Blackbody Radiation Thermal Imaging, Thermal Properties.

Case Study: Understanding an image and visual interpretation using SAGA.

UNIT III: Image Processing (9 hrs)

Digital Image Processing: Introduction, Pre-processing - Preprocessing -Geometric Correction Methods, Radiometric Correction Methods, Atmospheric Correction Methods, Image Enhancement- Contrast Enhancement, Spatial Filtering Techniques- Low Pass Filters, High Pass Filters, Filtering for Edge Enhancement, Image Transformations - NDVI Transformation, PCA Transformation.

UNIT IV: Image Classification (7 hrs)

Image Classification - Image Classification, Supervised Classification, Training Dataset Unsupervised Classification, Landuse/Landcover Classification System using Maximum Likelihood Classifier, Classification Accuracy Assessment, Post-classification Processing.

.

UNIT V: Applications (8 hrs)

Applications of Remote Sensing: Introduction, Land Cover and Land Use, Land-use/Land-cover Change, Land-cover Mapping, Agriculture, Crop Type Mapping, Crop Monitoring and Crop Damage Assessment, Forestry, Clear-cut Mapping and Deforestation, Species Identification and Typing, Burn Mapping.

Text Book(s):

[1]. Remote sensing and GIS, Bhatta B (2008), Oxford University Press

[2].Remote Sensing and Geographic Information System, Anji Reddy, Third Edition BS

Reference Books:

[1] Lillesand T.M. and Kiefer R.W., *Remote Sensing and Image Interpretation* (4th ed.), John Wiley and Sons Inc., New York, 2000

[2] Fundamentals of Remote Sensing – George Joseph, University Press

[3] Applied Remote Sensing, C.P. Lo, Longman Scientific and Technical Publishers

E-resources and other digital material

[1]. Prof. Pennan Chinnasamy, Centre for Technology Alternatives for Rural Areas (CTARA) Department,IIT Bombay https://nptel.ac.in/courses/106/105/106105077/

[2]. Don Boyes, Professor, Geography and Planning https://in.coursera.org/specializations/gis-mapping-spatial-analysis#instructors

[3]. Indian Geospatial education portal https://dst-iget.in/

AGILE AND SCRUM METHODOLOGIES

Course Code	24IT310D		
Course Category:	Professional Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	24IT205 Software Engineering	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course provides a comprehensive understanding of Agile software development methodologies with emphasis on Scrum and Extreme Programming (XP). It introduces Agile values, principles, Scrum roles, events, and artifacts used in iterative software development. The course focuses on customer collaboration, teamwork, time management, Agile estimation techniques, quality assurance, and project management practices. Students learn concepts such as user stories, sprint planning, test-driven development, trust, learning, and abstraction in Agile environments. The course also explores XP practices and strategies for embracing change to deliver high-quality software efficiently in dynamic development environments.

COURSE OBJECTIVES

1. Understand the principles, values, and practices of Agile software development and Scrum framework.
2. Explain the roles of customers, users, Agile teams, and effective collaboration in software projects.
3. Apply Agile estimation, quality assurance, and project management techniques in iterative development environments.
4. Analyze teamwork, learning, trust, abstraction, and time management practices in Agile development.
5. Develop the ability to adopt Extreme Programming (XP) practices and embrace change in software engineering projects.

COURSE OUTCOMES

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

CO1: Summarize Agile values, principles, Scrum framework, roles, events, and artifacts used in Agile software development. [K2]

CO2: Illustrate customer collaboration, user stories, teamwork, and time management practices in Agile environments. [K2]

CO3: Apply Agile estimation techniques, sprint planning, backlog refinement, and quality assurance practices in software projects. [K3]

CO4: Analyze learning, abstraction, trust, ethics, and transparency concepts in Agile software development processes. [K4]

CO5: Develop software solutions using Extreme Programming (XP) practices while adapting to changing project requirements effectively. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Introduction to Agile Software Development (8 hrs)

Learning Agile: Agile, Understanding Agile values, The Agile Manifesto, Three Perspectives on Software Engineering, The 12 Principles of Agile Software, Applications of Agile Software Development

Scrum Theory: Transparency, Inspection, Adaptation

Scrum Values

Scrum Team: Developers, Product Owner, Scrum Master

Scrum Events: The Sprint, Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective

Scrum Artifacts: Product Backlog, Sprint Backlog, Increment

Introduction to Kanban: Agile Software Development approach-Kanban, The Principles of Kanban, Improving Your Process with Kanban

UNIT II: **Customer Collaboration, Teamwork and Time Management in Agile (9 hrs)**

Customers and Users: Overview, Objectives**, Customer -** The Customer role, Customer Collaboration, The Customer Is Always Right, The User – User Objectives, User Stories, Combining UCD with Agile Development, Customers and Users in Learning Environments

Acceptance Criteria

Teamwork- Overview**,** Objectives, A Role Scheme in Agile Teams, Dilemmas in Teamwork, Teamwork in Learning Environments.

Time: Overview, Objectives, Time-Related Problems in Software Projects, Tightness of Software Development Methods, Sustainable Pace, Time Management of Agile Projects, Time in Learning Environments.

UNIT III: Story points & estimation techniques (8 hrs)

Agile estimation techniques: T-Shirt Sizing, Sprint Poker, Three-Point Method, Affinity Estimation, Relative Mass Evaluation, Dot voting, Maximum allowable size (MAS), Big, Uncertain, Small.

Case Study: Which Agile Estimation Technique Should Your Team Use?

Agile Measures- Objectives, Importance of Measures, Case Study- Monitoring a Large-Scale Project by Measures, Measures in Learning Environments

Quality- Overview, Objectives, The Agile Approach to Quality Assurance, Test-Driven Development, Quality in Learning Environments

UNIT IV:

Agile Project Management: (9 hrs)

Sprint planning & backlog refinement**,** Iterations, Business day

Learning: Overview, Objectives**,** How Does Agile Software Development Support Learning Processes, Learning in Learning Environments.

Abstraction: Overview, Objectives, Abstraction Levels in Agile Software Development, Abstraction in Learning Environments

Trust: Overview, Objectives, Software Intangibility and Process Transparency, Game Theory Perspective in Software Development, Ethics in Agile Teams, Diversity, Trust in Learning Environments

UNIT V: Extreme Programming (XP) and Embracing Change (8 hrs)

XP and Embracing Change: Going into Overtime, The Primary Practices of XP, The Game Plan Changed, but We're Still Losing, The XP Values Help the Team Change Their Mindset, An Effective Mindset Starts with the XP Values, The Momentum Shifts, Understanding the XP Principles Helps You Embrace Change.

Text Book(s):

[1]. Hazza and Dubinsky, ―Agile Software Engineering, Series: Undergraduate Topics in Computer Science, Springer, 2009.

[2]. Andrew Stellman, Jill Alison Hart, Learning Agile, O'Reilly, 2015.

[3]. Industrial Reference**:**
Scrum official guide: https://scrumguides.org/scrum-guide.html

[4]. Estimation techniques Reference: https://www.zstream.io/blog/top-7-most-popular-agile-estimation-techniques

Reference Books:

[1]. Andrew Stellman, Jennifer Green, Head First Agile, O'reilly, 2017..

[2]. Rubin K , Essential Scrum : A Practical Guide To The Most Popular Agile Process, Addison-Wesley, 2013

E-resources and other digital material

[1] Praveen Mittal, Adjunct Professor, University of Minnesota
https://www.coursera.org/learn/agile-software-development

[2] Prof. Rajib Mall, CSE IIT Kharagpur, NPTEL Course name: Agile model
https://archive.nptel.ac.in/courses/106/105/106105182/

[3] Nate Dinet, Author & Co-founder of Conquer Life & Enterprise Agile Coach, Course name: An overview of agile methodologies
https://www.udemy.com/course/agile-methodologies-overview/

[4] Bertrand Meyer : Agile methods are one of the most important developments
https://learning.edx.org/course/course-v1:ETHx+ASD.1x+2T2020/home

INTERNET OF THINGS : DESIGN AND IMPLEMENTATION

Course Code	**24IT310E**		
Course Category:	Professional Elective	**Credits:**	03
Course Type:	Theory	**Lecture - Tutorial - Practice:**	3-0-0
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	40 60 100

COURSE DESCRIPTION

This course introduces the fundamental concepts, architecture, communication protocols, and enabling technologies of the Internet of Things (IoT). It provides knowledge on IoT connectivity principles, sensors, RFID, wireless sensor networks, and networking protocols used in smart applications. The course also covers IoT system design methodologies, cloud-based data collection, storage, processing, and analytics techniques. Students gain understanding of IoT security, privacy, and vulnerability issues along with suitable protection mechanisms. Practical exposure to IoT devices such as Raspberry Pi and Python programming enables students to develop real-time IoT applications for various domains.

COURSE OBJECTIVES

1. Understand the evolution, architecture models, and enabling technologies of IoT.
2. Analyze IoT communication protocols, access technologies, and networking standards.
3. Program Raspberry Pi platforms to interface sensors, actuators, and cloud services.
4. Apply data acquisition, analytics, and cloud computing techniques for IoT applications.
5. Evaluate IoT security threats and design secure, scalable IoT systems aligned with standards.
6. Design IoT-based systems using networking protocols, IoT platforms, and suitable design methodologies for real-world applications.

COURSE OUTCOMES

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

CO1: Outline the fundamentals, architecture, enabling technologies, and deployment models of IoT systems. [K2]

CO2: Illustrate Internet connectivity principles, communication protocols, sensors, RFID, and wireless sensor networks used in IoT applications. [K2]

CO3: Apply cloud computing, data acquisition, storage, processing, and analytics techniques for IoT environments. [K3]

CO4: Analyze IoT networking protocols and design IoT solutions using appropriate methodologies for real-time applications. [K4]

CO5: Develop IoT-based applications using Raspberry Pi and Python while addressing IoT security and privacy challenges. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Fundamentals of IoT and Architecture (7 hrs)

Introduction to Internet of things: Introduction, Physical design of IoT, Logical Design of IoT,

IoT Enabling technologies, IoT levels & Deployment templates.

IoT Architecture: IoT Conceptual Framework, Architectural View, Technologies behind IoT, Sources of IoT, M2M Communication, IoT examples.

UNIT II: **IoT Communication Protocols (9 hrs)**

Internet Connectivity Principles: Introduction, Internet Connectivity, Internet-Based Communication, IP Addressing in the IoT, Media Access Control, Application Layer Protocols-HTTP, HTTPS, FTP, Telnet and others.

Sensors, Participatory Sensing, RFIDs and Wireless Sensor Networks: Introduction, Sensor Technology, Actuator, Sensor Data Communication Protocols, Radio Frequency Identification Technology.

UNIT III: Networking Protocols for IoT and Design Methodology (9 hrs)

Wireless Protocols and Technologies for Internet of Things - WPAN Technologies for IoT: IEEE 802.15.4, Zigbee, HART, NFC, Z-Wave, Bluetooth, BACnet, Modbus, Li-Gi, Wi-Fi, **IP-Based Networking Protocols for IoT –** Overview of Networking, TCP/IP, 6LoWPAN, RPL, REST, AMQP, XMPP, CoAP, MQTT, LoRa

IoT Platforms Design Methodology: Introduction, IoT Design Methodology,

Case Study on IoT System for Weather Monitoring

UNIT IV: IoT Data Collection and Storage (8 hrs)

Data Acquiring, Organizing, Processing and Analytics: Introduction, Data Acquiring and Storage, Organizing the Data, Transactions, Business, Processes, Integration and Enterprise Systems, Analytics.

Data Collection, Storage and Computing Using a Cloud Platform: Introduction, Cloud Computing Paradigm for Data Collection, Storage and Computing, Everything as a Service and Cloud Service Models

.

UNIT V: IoT Security and Application Development (8 hrs)

IoT Privacy, Security and Vulnerabilities Solutions: Vulnerabilities, Security Requirements and Threat Analysis, IoT Security Tomography and Layered Attacker Model

IoT physical devices & EndPoints: IoT Device, Raspberry Pi Board, Raspberry Pi interfaces, Programming Raspberry pi with python.

Text Book(s):

[1]. Raj Kamal, "Internet of Things: Architecture and Design Principles", 2nd Edition, McGraw Hill Education (India), 2022. (ISBN: 978-9390727384)

[2]. Vijay Madisetti and ArshdeepBahga, **"Internet of Things (A Hands- on-Approach)",** 1st Edition, University Press Private Limited,2017

[3]. Pramod R. Gunjal, Satish R. Jondhale, Jaime Lloret Mauri and Karishma Agrawal, "Internet of Things: Theory to Practice", 1st Edition, Routledge / CRC Press, 2024. (ISBN: 978-1032251684)

Reference Books:

[1]. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, 2017.

[2]. Shriram K. Vasudevan, Abhishek S. Nagarajan and R.M.D. Sundaram, "Internet of Things", John Wiley & Sons, 2023

[3]. Massimo Banzi and Michael Shiloh, "Getting Started with Arduino", 4th Edition, Maker Media, 2022.

E-resources and other digital material

[1]. Prof SudipMisra, IIT, Kharagpur, "Introduction to Internet of Things", NPTEL 2022
https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs96

[2]. Prof. Neil, Edureka, IoT Tutorial for Beginners | Internet of Things (IoT) | Edureka,2022
https://www.youtube.com/watch?v=UjN6wEfCLAo

[3]. Prof. Rajbabu Velmurugan, "Introduction to Internet of Things" Coursera 2025
Introduction to Internet of Things | Coursera

CLOUD COMPUTING LAB

Course Code	**24IT383**		
Course Category:	Program Core	**Credits:**	1.5
Course Type:	Laboratory	**Lecture - Tutorial - Practice:**	0-0-3
Prerequisites:	Computer Networks	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	60 40 100

COURSE DESCRIPTION

This laboratory course provides practical exposure to core concepts of cloud computing through hands-on experiments and real-time application development. Students explore web services, virtualization, inter-process communication, and distributed systems to understand cloud architectures. The course covers deployment and management of applications using major cloud platforms such as AWS, Google Cloud, and Salesforce, along with cloud storage, monitoring, and security services. Advanced topics including serverless computing and messaging systems are also introduced, enabling students to design, deploy, and manage scalable cloud-based solutions.

COURSE OBJECTIVES

1. Understand web services, inter-process communication, and distributed messaging systems to build foundational cloud communication models.
2. Introduce Virtualization concepts through installation and management of virtual machines.
3. Develop the ability to utilize cloud infrastructure services including compute, storage, monitoring, and security using platforms such as Amazon Web Services and Google Cloud.
4. Build Serverless applications using platforms such as OpenFaaS, Salesforce, and Google App Engine.
5. Apply cloud deployment concepts to deploy scalable real-time applications on public cloud platforms such as AWS, Azure, and GCP with appropriate security and performance considerations.

COURSE OUTCOMES

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

CO1: Implement Web services, inter-process communication models, and distributed messaging applications in cloud environments. (K3)

CO2: Configure virtual machines using virtualization platforms such as Oracle VM VirtualBox and VMware Workstation. (K3)

CO3: Utilize cloud infrastructure services including compute, storage, monitoring, and security using cloud platforms such as Amazon Web Services and Google Cloud. (K3)

CO4: Develop cloud-native and serverless applications using platforms such as OpenFaaS, Salesforce, and Google App Engine. (K3)

CO5: Deploy and manage real-time applications on cloud platforms such as AWS, Microsoft Azure, or Google Cloud Platform (K6)

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES

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

COURSE CONTENT

Week 1:

Implement basic Web Services

Develop and test simple REST/SOAP-based web services to understand service-oriented communication in cloud environments.

Week 2:

Work with Inter-Process Communication (IPC) and Messaging Systems

Configure publish/subscribe models and message queues to demonstrate distributed communication.

Week 3:

Install and Configure Virtual Machines

Set up VirtualBox/VMware Workstation and deploy multiple Linux/Windows OS instances on a host system.

Week 4:

Program Development and Execution in a Virtual Environment

Install a C compiler inside a virtual machine and execute basic programs to observe compilation and execution inside a VM.

Week 5:

Launch and Configure an Amazon EC2 Instance

Create an EC2 virtual server, deploy a web server, and configure security groups to allow HTTP access.

Week 6:

Application Hosting using Google App Engine

Create and deploy simple web applications using Python/Java on Google App Engine.

Week 7:

File Transfer Across Virtual Machines

Demonstrate secure file transfer between Virtual Machines

Week 8:

Cloud Storage Services

Create and manage cloud storage resources using Amazon S3 by performing data operations with access control and lifecycle policies.

Week 9:

Cloud Monitoring and Logging
Configure and use Amazon CloudWatch to collect metrics, analyze logs, and set alerts for application performance.
Week 10:
Serverless Computing using OpenFaaS
Deploy and invoke serverless functions using OpenFaaS, demonstrating event-driven execution.
Week 11:
Salesforce Cloud Platform for pplication Development
Create a Salesforce Developer Account, build a simple application using Objects, Apex Classes, Triggers, and Lightning App Builder.
Demonstrate cloud-native CRM development and deployment.
Week 12: Project
Deploying Applications to the Cloud (AWS / Azure / GCP)
Package and deploy a real-time or web application to a cloud platform using any of the following:

- AWS Elastic Beanstalk
- Azure App Services
- Google Cloud Run / Cloud Functions.

Text Book(s):

[1]. Mastering Cloud Computing, 2nd edition, Rajkumar Buyya, Christian Vecchiola,Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, Mc Graw Hill, 2024.
[2]. Velte T. Antony, Velte J. Toby., Elsenpeter Robert, "Cloud Computing: A Practical Approach", Tata McGraw- Hill, 2010.

Reference Books:

[1]. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
[2]. Essentials of cloud Computing, K. Chandrasekhran, CRC press, 2014.

E-resources and other digital material

[1]. Prof. Soumya Kanti Ghosh, IIT Kharagpur, "Cloud Computing", 2025, https://nptel.ac.in/courses/106105167.
[2]. Udemy – "Cloud Computing: The Complete Guide for Beginners", 2025, https://www.udemy.com/course/intro-to-cloud-computing/.
[3]. AWS Training & Certification – Cloud Essentials, Serverless & Security, 2024, https://aws.amazon.com/training.
[4]. Microsoft Learn – Azure Cloud Fundamentals, 2024, https://learn.microsoft.com/en-us/training.
[5]. Google Cloud Skills Boost – Cloud Engineering & Serverless Development, 2024, https://www.cloudskillsboost.google.

ADVANCED JAVA PROGRAMMING LAB

Course Code:	**24IT384**		
Course Category:	Program Core	**Credits:**	1.5
Course Type:	Laboratory	**Lecture - Tutorial - Practice:**	0-0-3
Prerequisites:	Java Programming	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	60 40 100

COURSE DESCRIPTION

This laboratory course provides practical exposure to advanced Java programming concepts for building enterprise-level and web-based applications. The course focuses on Java Database Connectivity (JDBC), Java Servlets, session management, and modern application development using Spring Boot. Students gain hands-on experience in database connectivity, server-side programming, dependency injection, inversion of control, annotations, and relational database integration using JDBC and JPA. The laboratory also introduces real-world application development practices through Spring Boot–based solutions and case studies, enabling students to design scalable, maintainable, and database-driven applications.

COURSE OBJECTIVES

1. Understand database connectivity concepts using JDBC and relational databases.
2. Develop server-side web applications using Java Servlets and session tracking mechanisms.
3. Design dynamic web applications that interact with relational databases.
4. Introduce Spring Boot concepts including dependency injection and inversion of control.
5. Develop enterprise applications using Spring Boot annotations, JDBC, and JPA. Apply advanced Java programming concepts to solve real-world application development problems.

COURSE OUTCOMES

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

CO1: Implement Java Database Connectivity Application Programming Interface to connect to relational databases. (K3)

CO2: Build server-side applications to interact with servers using Java Servlets. (K3)

CO3: Design web applications that interact with servers as well as relational databases. (K3)

CO4: Implement dependency injection and inversion of control to solve problems in Spring Boot. (K3)

CO5: Apply Spring Boot annotations to provide solutions to real-world problems. (K3)

CO6: Create Spring Boot applications using Spring Boot annotations and database integration. (K6)

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2
CO1		2			2									
CO2	3				2									
CO3			3		2								2	2
CO4	2				3								1	
CO5	3				2								2	2
CO6			3		3								2	2

COURSE CONTENT

Week 1:

Create JDBC Programs to Access Databases

Develop JDBC applications to connect Java programs with relational databases such as:
a. MS Access
b. MySQL

Week 2:

Database Connectivity using Statement Interfaces and ResultSet

Use Statement, Prepared Statement, and Callable Statement interfaces to process database operations and implement CRUD operations using ResultSet.

Week 3:

Create Server-Side Applications using Java Servlets

Develop and execute Java Servlet programs to handle client requests and generate server responses.

Week 4:

Servlet Programs on Session Tracking

Implement session management techniques using:
a.Cookies
b. Sessions

Week 5:

Creation of Spring Programs using Spring Boot

Create Spring Boot applications using:

- SpringBootApplication
- Spring Initializer

Week 6:

Implementation of 12-Factor Applications in Spring Boot

Develop scalable and configurable applications following 12-factor application principles in Spring Boot.

Week 7:

Dependency Injection in Spring Boot

Implement dependency injection and inversion of control concepts in Spring Boot applications.

Week 8:

Use of Annotations in Spring Boot

Develop applications using Spring Boot annotations such as:

- @Component
- @Autowired
- @Service
- @Repository
- @Controller

Week 9:

Database Access using JDBC in Spring Boot

Access relational databases using JDBC integration in Spring Boot applications.

Week 10:

Relational Database Access using JPA

Implement relational database operations using Java Persistence API (JPA) and Spring Data JPA.

Week 11:

Case Study – Web Applications using Java Servlets

Develop dynamic web applications using Java Servlets with database connectivity and form handling.

Week 12:

Case Study – Spring Boot Applications for Real-World Problems

Develop Spring Boot applications for real-world scenarios involving:

1. Session handling
2. Database integration
3. REST-based web applications

Text Book(s)

[1]. Keogh J. *J2EE: The Complete Reference*. 1st ed. New York: McGraw-Hill Education; 2002.
[2]. Bakliwal S. *Hands-on Application Development using Spring Boot*. 1st ed. New Delhi: BPB Publications; 2022.

Reference Book(s)

[1]. Walls C. *Spring in Action*. 6th ed. Shelter Island: Manning Publications; 2021.
[2]. Heckler M. *Spring Boot: Up and Running*. Sebastopol: O'Reilly Media; 2021.

E-Resources and Other Digital Material

[1]. Karanam R. *Java Servlets and JSP - Build Java EE (JEE) App in 25 Steps* [Internet]. Udemy; 2022 [cited 2026 May 12]. Available from: https://www.udemy.com/course/learn-java-servlets-and-jsp-web-application-in-25-steps/
[2]. Spring Boot Official Documentation [Internet]. Spring.io; 2022 [cited 2026 May 12]. Available from: https://spring.io/projects/spring-boot
[3]. *Advanced Java Programming* [Internet]. Udemy; 2022 [cited 2026 May 12]. Available from: https://www.udemy.com/advanced-java-programming/
[4]. Parsons D. *Spring MVC, Spring Boot and REST Controllers* [Internet]. Coursera; 2022 [cited 2026 May 12]. Available from: https://www.coursera.org/learn/spring-mvc-rest-controller
[5]. Karanam R. *Spring Framework Master Class - Java Spring the Modern Way* [Internet]. Udemy; 2022 [cited 2026 May 12]. Available from: https://www.udemy.com/course/spring-tutorial-for-beginners/

ADVANCED ENGLISH COMMUNICATION SKILLS LAB

Course Code:	**24EN381**		
Course Category:	HS	**Credits:**	1
Course Type:	Laboratory	**Lecture - Tutorial - Practice:**	0-0-2
Prerequisites:	Intermediate English	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	100 - 100

COURSE DESCRIPTION

Advanced English Communication Skills Lab course covers a range of essential communication skills for engineering students. Topics include Vocabulary Building, Listening Comprehension with TED Talks, effective Information Transfer, Technical Report Writing, Book Review, Resume Writing, Video Resume Creation, Interview Skills, Organizing Meetings & Multitasking, Nonverbal Communication, Group Discussions, and Presentation Skills, all aimed at enhancing proficiency in both written and verbal communication..

COURSE OBJECTIVES

1. Explain the fundamentals of language, including vocabulary and listening comprehension, relevant to professional environments.
2. Familiarize students with written communication required for administrative and corporate documentation.
3. Develop spoken communication skills, including argumentation, public speaking, and presentation techniques.
4. Teach the principles of nonverbal communication and their effective use in professional and social contexts with clarity and conciseness.
5. Instruct students in meeting organization practices and multitasking strategies to enhance productivity in professional settings.

COURSE OUTCOMES

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

CO1: Identify and recall the fundamentals of language, including vocabulary and listening comprehension, relevant to professional environments [K1].
CO2: Explain written communication practices required for administrative and corporate documentation [K2].
CO3: Apply spoken communication skills, including argumentation, public speaking, and presentation techniques, in professional contexts [K3].
CO4: Analyze the role of nonverbal communication in achieving clarity and conciseness in professional and social interactions [K4].
CO5: Evaluate meeting organization practices and multitasking strategies to enhance productivity in professional settings [K5].

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PSO 1	PSO 2
CO 1	2								3				
CO 2									3				
CO 3									3				
CO 4									3				
CO 5								2	3		2		

COURSE CONTENT

Laboratory Exercises

1) Vocabulary: Vocabulary Building – Business Vocabulary – Idioms – Antonyms – Synonyms – Analogies. (2 Hours)

Description:

The Vocabulary unit focuses on building vocabulary skills, including business vocabulary, idioms, antonyms, synonyms, and analogies, to enhance language proficiency and effective communication in professional settings.

Applications:

1) Walden
2) Softx

Exercises:

1) Business Vocabulary
2) Idioms
3) Synonyms & Antonyms
4) Analogies

Specific Resources:

AI Tools:

Quizlet
https://quizlet.com/qchat-personal-ai-tutor

Anki
https://www.brainscape.com/subjects/vocab-builder

2) Book Review: Elements of a Book Review (Summary, Analysis, Critique) – Writing Style and Tone in Book Reviews – Evaluating the Strengths and Weaknesses of a Book – Personal Reflections and Recommendation. (2 Hours)

Description:

The Book Review unit focuses on the elements of a book review (summary, analysis, critique), writing style and tone, evaluating strengths and weaknesses, and providing personal reflections and recommendations.

Examples:
1) Walden
2) Softx
Exercises:
1) "The Immortals of Meluha" by Amish Tripathi
2) "The God of Small Things" by Arundhati Roy
3) "Train to Pakistan" by Khushwant Singh
4) "The White Tiger" by Aravind Adiga
5) "A Suitable Boy" by Vikram Seth
6) "The Palace of Illusions" by Chitra Banerjee Divakaruni
7) "The Guide" by R.K. Narayan
8) "The Great Indian Novel" by Shashi Tharoor
9) "Interpreter of Maladies" by Jhumpa Lahiri
10) "Midnight's Children" by Salman Rushdie
Specific Resources:
AI Tools:
Writesonic
https://writesonic.com/
Quillbot
https://quillbot.com/
AI Writer
https://ai-writer.com/

3) Listening Comprehension: Ted Talks – Active Listening Strategies – Note-taking Techniques –Summarizing and Analysing Ted Talks. (2 Hours)
Description:
The Listening Comprehension unit includes TED Talks, active listening strategies, note-taking techniques, and skills for summarizing and analyzing TED Talks, enhancing students' understanding and retention of spoken content.
Applications:
1) Walden
2) Softx
Exercises:
1) Have students listen to a Ted Talk on a relevant engineering topic. Afterwards, ask them to write a summary of the main points and key ideas presented in the talk.
2) Provide students with a short Ted Talk audio clip. Ask them to listen carefully and take notes on the main ideas, supporting details, and any important facts or figures mentioned.
3) After listening to a Ted Talk, provide students with a set of discussion questions related to the content. Encourage them to discuss their answers and opinions with classmates.
4) Select a Ted Talk that presents a problem or challenge related to engineering. Ask students to listen and analyse how the speaker presents the issue, the solutions proposed, and the overall impact of the talk.
5) Play a Ted Talk audio clip and pause at various points to ask students to summarize what they have heard so far. This encourages active listening and comprehension.

6) Have students pair up and listen to each other's summaries of a Ted Talk. They can provide feedback on the clarity, accuracy, and completeness of the summaries.
7) Choose two Ted Talks on similar engineering topics but with different perspectives. Have students listen to both talks and compare and contrast the approaches taken by the speakers.
Specific Resources:
AI Tools:
Duolingo
https://www.duolingo.com/ Cambly
https://www.cambly.com/english/resources

4) Nonverbal Communication: Importance of nonverbal Communication – Kinesics – Proxemics – Haptics – Oculesics – Chronemics – Olfactics – Vocalics – Sound Symbols and Signs – Silence – Posture – Adornment – Locomotion – Communication by posture, charts, maps and graphs – Communication by colours – Communication by human behaviour. (2 Hours)

Description:
The Nonverbal Communication unit explores the importance of nonverbal communication, including kinesics, proxemics, haptics, oculesics, chronemics, olfactics, vocalics, sound symbols and signs, silence, posture, adornment, locomotion, communication through posture, charts, maps, graphs, colors, and human behavior.
Applications:
1) Walden
2) Softx
Exercises:
1) Discuss the importance of nonverbal communication in engineering contexts. How can engineers use nonverbal cues to enhance their communication skills?
2) Explain the concept of kinesics in nonverbal communication. How can engineers use body language to convey messages effectively?
3) Describe the role of proxemics in engineering communication. How does the use of space impact the way engineers interact with others?
4) How can engineers use haptics to communicate information effectively? Provide examples of how touch can be used in engineering contexts.
5) Explain the significance of oculesics in nonverbal communication. How can engineers use eye contact to establish rapport and convey interest?
6) Discuss the role of chronemics in engineering communication. How does the perception of time impact communication in engineering projects?
7) Describe the use of olfactics in engineering communication. How can engineers use scent to create a positive work environment?
8) Explain the concept of vocalics in nonverbal communication. How can engineers use tone, pitch, and volume to convey meaning?
9) Discuss the use of sound symbols and signs in engineering communication. How can engineers use sound to communicate complex ideas?
10) Describe how engineers can use nonverbal cues such as silence, posture, adornment, and locomotion to enhance their communication skills in engineering projects.
Specific Resources:
AI Tools:

Cogito
https://cogitocorp.com/
Affectiva
https://www.affectiva.com/
Receptiviti
https://www.receptiviti.com/

5) Presentations: Planning and Structuring a Presentation – Effective Use of Visual Aids – Public Speaking Techniques – Engaging the Audience – Handling Q&A Sessions. (2 Hours)

Description:

The Presentations unit focuses on planning and structuring presentations, effective use of visual aids, public speaking techniques, engaging the audience, and strategies for handling Q&A sessions, enhancing students' presentation skills.

Applications:

1) Walden
2) Softx

Exercises:

1) Sustainable transportation solutions for urban areas.
2) Advancements in renewable energy sources: Solar, wind, and hydroelectric power.
3) Innovative approaches to reduce air pollution in cities.
4) The environmental impact of plastic waste and solutions for reduction.
5) Say no to single-use plastics: Alternatives and their benefits.
6) Recycling waste: Technologies and processes for efficient waste management.
7) Green buildings and sustainable architecture: Design principles and benefits.
8) The role of engineering in water conservation and management.
9) Smart cities: Technologies for sustainable urban development.
10) Sustainable agriculture practices: Engineering solutions for food security.

Specific Resources:

AI Tools:

Visme
https://www.visme.co/
Emaze
https://www.emaze.com/
Prezi
https://prezi.com/

6) Interview Skills: Preparing for Interviews – Types of Interviews (Behavioral, Technical, etc.) – Common Interview Questions and Answers – Body Language and Non-verbal Communication in Interviews – Follow-up and Thank You Notes. (2 Hours)

Description:

The Interview Skills unit covers interview preparation, types of interviews (behavioral, technical, etc.), common questions and answers, body language and non-verbal communication, and follow-up and thank you notes.

Applications:

1) Walden

2) Softx
Exercises:
1) Tell me about yourself.
2) What are your strengths?
3) What are your weaknesses?
4) Can you describe a challenging situation you faced at work and how you handled it?
5) How do you handle stress and pressure?
6) Tell me about a time when you had to work with a difficult colleague or team member. How did you handle it?
7) Why do you want to work for our company?
8) Where do you see yourself in five years?
9) Can you give an example of a time when you demonstrated leadership skills?
10) How do you handle constructive criticism?
11) Describe a time when you had to multitask or manage multiple projects simultaneously.
12) Can you give an example of a time when you had to adapt to a change in a work environment?
13) How do you prioritize your work and manage your time effectively?
14) Describe a situation where you had to resolve a conflict with a customer or client.
15) How do you stay motivated in your work?
16) Can you give an example of a time when you had to meet a tight deadline?
17) Describe a time when you had to take initiative to solve a problem.
18) How do you handle failure or setbacks?
19) Can you describe a time when you had to work on a team project?
20) What motivates you to succeed?
Specific Resources:
AI Tools:
Big Interview
https://www.biginterview.com/
MockQuestions
https://www.mockquestions.com/
MyInterview
https://myinterview.com/
HireVue
https://www.hirevue.com/

7) Group Discussion: Purpose and Importance of Group Discussions – Strategies for Effective Participation – Active Listening Techniques – Constructive Feedback Giving and Receiving – Handling Conflicts in Group Discussions – Mock GDS. (2 Hours)
Description:
The Group Discussion unit covers the purpose and importance of group discussions, strategies for effective participation, active listening techniques, constructive feedback giving and receiving, handling conflicts, and includes mock group discussion sessions for practice.
Applications:
1) Walden
2) Softx

Exercises:

1) The impact of artificial intelligence on engineering design and innovation.
2) Sustainable engineering practices for a greener future.
3) The role of blockchain technology in transforming engineering industries.
4) Advancements in renewable energy technologies and their potential applications.
5) Challenges and opportunities in implementing 5G technology in engineering.
6) The future of transportation: Electric vehicles vs. hydrogen fuel cell vehicles.
7) The integration of Internet of Things (IoT) in smart cities and infrastructure development.
8) Ethical considerations in engineering, particularly in areas like AI and biotechnology.
9) The role of engineering in addressing global climate change and environmental sustainability.
10) Innovations in robotics and automation and their impact on the workforce.

Specific Resources:

AI Tools:

Glider AI

https://glider.ai/

Youth4Work

https://www.youth4work.com/

8) Resume Writing: Components of a Resume (Contact Information, Summary, Experience, Education, Skills, etc.) – Tailoring the Resume for Different Job Applications – Formatting and Layout Guidelines – Keywords and ATS (Applicant Tracking System) Optimization. (2 Hours)

Description:

The Resume Writing unit covers resume components (contact information, summary, experience, education, skills), tailoring for different job applications, formatting and layout guidelines, and optimizing keywords for ATS (Applicant Tracking System).

Applications:

1) Walden
2) Softx

Exercises:

1) We are excited to announce that there was an advertisement in the Hindu daily newspaper on 22-06-2024 for the position of Junior Software Engineer. If you are passionate about coding, problem-solving, and creating innovative software solutions, we encourage you to apply! Please submit your resume and a well-crafted cover letter highlighting your skills, experiences, and enthusiasm for software development.
2) We are excited to announce that there was an advertisement in The Hindu daily newspaper on 15-05-2024 for the position of Senior Electrical Engineer. If you are passionate about electrical systems, innovative designs, and sustainable engineering solutions, we encourage you to apply! Please submit your resume and a well-crafted cover letter highlighting your skills, experiences, and enthusiasm for electrical engineering.
3) We are excited to announce that there was an advertisement in The Hindu daily newspaper on 01-06-2024 for the position of Mechanical Design Engineer. If you are passionate about mechanical systems, innovative design, and precision engineering, we encourage you to apply! Please submit your resume and a well-crafted cover letter highlighting your skills, experiences, and enthusiasm for mechanical engineering.

4) We are excited to announce that there was an advertisement in The Hindu daily newspaper on 10-05-2024 for the position of Civil Project Manager. If you are passionate about civil engineering, project management, and sustainable construction practices, we encourage you to apply! Please submit your resume and a well-crafted cover letter highlighting your skills, experiences, and enthusiasm for civil engineering.
5) We are excited to announce that there was an advertisement in The Hindu daily newspaper on 25-06-2024 for the position of Network Security Specialist. If you are passionate about cybersecurity, network infrastructure, and safeguarding digital assets, we encourage you to apply! Please submit your resume and a well-crafted cover letter highlighting your skills, experiences, and enthusiasm for network security.
Specific Resources:
AI Tools:
Resume Genius
https://resumegenius.com/
Kickresume
https://www.kickresume.com/en/
Resumake
https://resumake.io/
9) Video Resume: Purpose and Benefits of a Video Resume –Scripting and Planning the Video Resume Filming and Editing Techniques – Showcasing Skills and Personality on Camera. (2 Hours)
Description:
The Video Resume unit explores the purpose and benefits of video resumes, scripting and planning, filming and editing techniques, and showcasing skills and personality on camera.
Applications:
1) Walden
2) Softx
Exercises:
1) We are excited to announce that there was an opening for the position of Junior Software Engineer. We encourage you to apply! Please submit a video resume. The video should be 2-3 minutes long and cover your key skills, relevant experiences, and what makes you a suitable candidate for this position. How have you ensured that your video resume effectively showcases your qualifications and passion for the Junior Software Engineer role?
2) We are happy to announce that there was an opening for the position of Data Analyst. We encourage you to apply! Please submit a video resume. The video should be 2-3 minutes long and cover your key skills, relevant experiences, and what makes you a suitable candidate for this position. How have you ensured that your video resume effectively showcases your qualifications and passion for the Data Analyst role?
3) We are enthusiastic to announce that there was an opening for the position of Senior Mechanical Engineer. We encourage you to apply! Please submit a video resume. The video should be 2-3 minutes long and cover your key skills, relevant experiences, and what makes you a suitable candidate for this position. How have you ensured that your video resume effectively showcases your qualifications and passion for the Senior Mechanical Engineer role?
4) We are pleased to announce an opportunity for the role of Lead Software Developer. We invite you to apply! Please submit a video resume. The video should be 2-3 minutes in duration and highlight your key skills, relevant experiences, and what makes you a strong candidate for

this position. How have you ensured that your video resume effectively showcases your qualifications and enthusiasm for the Lead Software Developer role?
5) We are thrilled to unveil an exciting opening for the coveted position of Principal Artificial Intelligence Architect. We cordially invite you to submit your application! Please prepare a video resume that succinctly captures your professional journey, showcasing your unparalleled expertise, innovative approach, and profound passion for pioneering AI solutions. How have you meticulously crafted your video resume to articulate your qualifications and fervor for the Principal Artificial Intelligence Architect role?"
Specific Resources:
AI Tools:
Veed.io
https://www.veed.io/
Animaker
https://www.animaker.com/
Biteable
https://biteable.com/

10) Technical Report Writing: Purpose and Audience Analysis–Structure of Technical Reports (Abstract, Introduction, Methodology, Results, Conclusion, Recommendations) – Clarity and Conciseness in Technical Writing – Citations and References. (2 Hours)
Description:
The Technical Report Writing unit covers purpose and audience analysis, structuring reports (abstract, introduction, methodology, results, conclusion, recommendations), clarity and conciseness, and proper use of citations and references.
Applications:
1) Walden
2) Softx
Exercises:
1) Write a Report on the feasibility of establishing a computer training institute in Tirupathi.
2) Write a Report on an Industrial Accident in Lanco Industrial Gases (Pvt.) Limited, Tirupathi.
3) Write a Report upon the desirability of promoting separate professional college for women.
4) Write a Feasibility Report upon introducing cooking gas in place of petrol and diesel in Automobiles.
5) You have been asked by a firm which manufactures detergent powder to make a study of the consumer reaction to their product and suggest measures to improve the image and the sales of their product. Prepare a report of the study.
Specific Resources:
AI Tools:
Jasper.ai
https://www.jasper.ai/
HyperWrite
https://www.hyperwriteai.com/
Writesonic
https://writesonic.com/

11) Information Transfer: Effective Communication of Information – Structuring Information for Clarity – Tailoring Information for Different Audiences – Verbal and Written Information Transfer Techniques. (2 Hours)

Description:

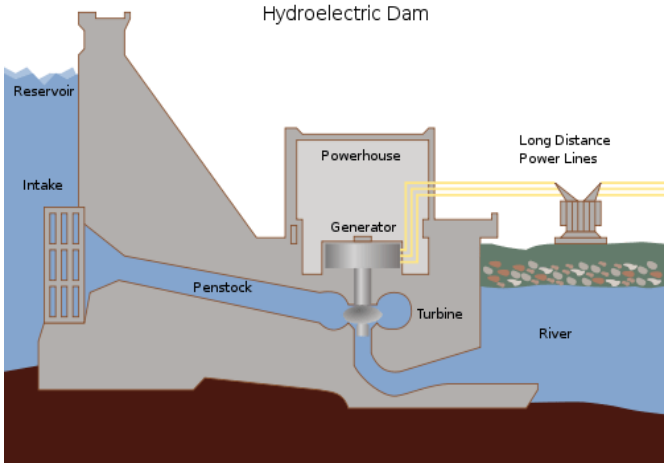
The Information Transfer unit covers effective communication, structuring information for clarity, tailoring information for different audiences, and mastering both verbal and written information transfer techniques.

Applications:

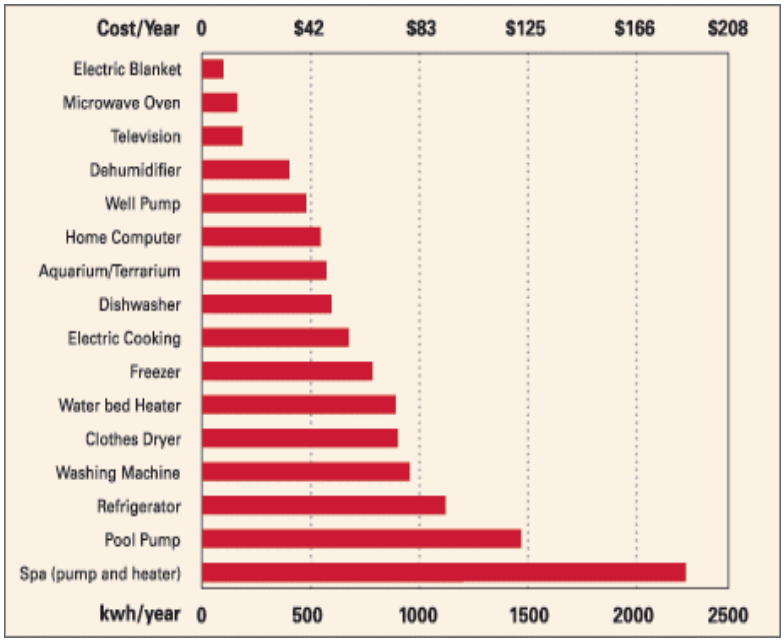
1) Walden
2) Softx

Exercises:

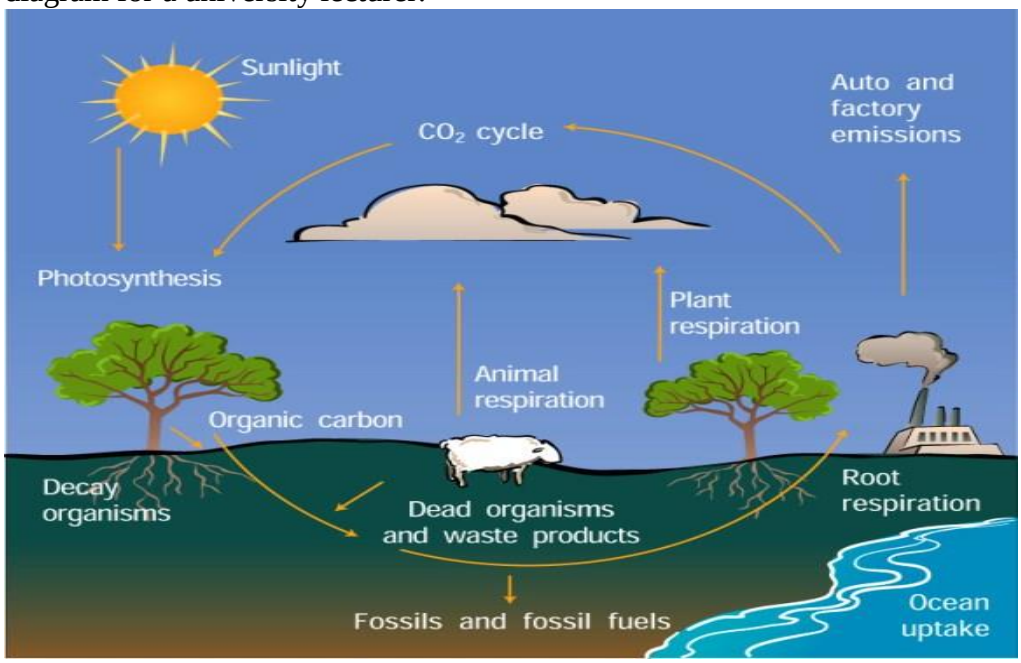
1) The diagram shows how electricity is generated by a hydroelectric dam. Write a 150-word report for a university lecturer explaining how the process works.



2) The bar chart shows the relative electricity consumption and cost per year of various household devices. Write a 150-word report for a university lecturer explaining the data and making comparisons where relevant.



3) The diagram below illustrates the carbon cycle in nature. Write a 150-word description of this diagram for a university lecturer.



Specific Resources:

AI Tools:

Gliffy

https://www.gliffy.com/

SmartDraw

https://www.smartdraw.com/

Tableau

https://www.tableau.com/

12) Organising Meetings & Multi-Tasking: Planning and Scheduling Meetings – Setting Agendas and Objectives – Facilitating Discussions and Decision Making – Time Management Techniques for Multitasking.
(2 Hours)
Description:
Organising Meetings & Multitasking unit includes planning and scheduling meetings, setting agendas and objectives, facilitating discussions and decision-making, and time management techniques for effective multitasking in professional settings.
Applications:
1) Walden
2) Softx
Exercises:
1) Explain the importance of organizing meetings in engineering project management. How does effective planning and scheduling contribute to the success of a meeting?
2) Discuss the key components of setting agendas and objectives for engineering meetings. How can a well-defined agenda help in achieving meeting goals?
3) Describe the role of a facilitator in engineering meetings. How can a facilitator ensure productive discussions and decision-making processes?
4) What are some time management techniques that engineering students can use to effectively multitask? How can these techniques improve productivity?
5) How can engineers prioritize tasks and manage their time effectively in a fast-paced work environment? Provide examples from engineering projects or experiences.
Specific Resources:
AI Tools:
Google Meet
https://meet.google.com/?pli=1
Zoom
https://zoom.us/
Microsoft Teams
https://www.microsoft.com/en-in/microsoft-teams/join-a-meeting

Text BookS:

[1]. Advanced English Communication Skills Lab Manual

References:

[1]. Chauhan, Gajendra Singh and SmithaKashiramka. (2018). Technical Communication. Cengage, Delhi.
[2]. Sanjay Kumar and PushpLata. (2011). Communication Skills. Oxford University Press, New Delhi.

English Language Communication Skills Lab Software:

1) Walden
2) Softx
3) Visionet Spears Digital Language Lab software Advance Pro
4) ODll Language Learner’s Software, Orell Techno Systems

VERSION CONTROL AND CI/CD PIPELINES

Course Code	24IT382A		
Course Category:	Skill Course	**Credits:**	1
Course Type:	Practical	**Lecture - Tutorial - Practice:**	0-0-2
Prerequisites:	-	**Continuous Evaluation:** **Semester-end Evaluation:** **Total Marks:**	100 - 100

COURSE DESCRIPTION

This course provides a practical approach to building distributed applications using containerization and automated pipelines. It focuses on the fundamental concepts of Docker, including container isolation, image creation, and persistent storage. It further covers version control integration with Git and GitHub to automate builds and deployments. Students will learn to orchestrate multi-container environments and implement Continuous Integration/Continuous Delivery (CI/CD) workflows to streamline software distribution.

COURSE OBJECTIVES

1. Understand the core architecture of Docker and the mechanics of container isolation.
2. Master the creation of optimized Docker images using Dockerfiles and version control.
3. Manage persistent data and complex networking between containers.
4. Implement multi-container orchestration using Docker Compose and cluster management.
5. Design and deploy automated CI/CD pipelines using Jenkins, Docker Hub, and registries.

COURSE OUTCOMES

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

CO1: Summarize the fundamentals of containerization and compare container isolation namespaces [K2]

CO2: Develop custom Docker images and implement versioning strategies using Git and tagging [K3]

CO3: Apply persistent storage solutions and network exposure archetypes to distributed systems [K3]

CO4: Explore multi-service environments by coordinating services, automating deployments, and scaling applications using Docker Compose and Docker Swarm [K3]

CO5: Develop automated CI/CD pipelines for consistent software delivery, integration, testing, deployment, and distribution [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PSO 1	PSO 2
CO 1	2	1	1									1	1
CO 2		1	1		2							2	2
CO 3		1	2	1	1							1	1
CO 4	1			1	2							1	2
CO 5	1		1		2							2	2

COURSE CONTENT

UNIT I: Introduction to Docker (6 hrs)

Introduction to Docker and Containers: Introduction, Containers vs Virtualization, Docker Architecture: Daemon and CLI.

Isolation Mechanisms: Linux Namespaces (PID, NET, MNT, IPC, UTS, USR), Resource protection with cgroups, Chroot.

Getting Started: Installing Docker on Ubuntu/CentOS, running "Hello World", Container Life cycle: Create, Start, Stop, and Remove.

Experiments:

1. **Installation and Verification of Docker Environment**
   Install Docker Engine on Ubuntu/CentOS Linux system and verify the installation by executing basic Docker commands. Create and run the “Hello World” container, inspect its details, and observe the Docker daemon and CLI interaction. Document the steps involved in container creation, execution, stopping, and removal.
2. **Exploring Container Isolation Mechanisms**
   Create multiple Docker containers and analyze process isolation using Linux namespaces such as PID, NET, IPC, and MNT. Observe container-specific process IDs, hostname separation, and file system isolation. Demonstrate resource limitations using cgroups by restricting memory and CPU usage for selected containers.
3. **Managing the Container Lifecycle**
   Develop a simple workflow to demonstrate the complete Docker container lifecycle. Create containers from existing images, start and stop containers, restart paused containers, and remove unused containers and images. Compare the behavior of containers with traditional virtual machines in terms of startup time and resource utilization.

UNIT II: Version Control and Image Creation (6 hrs)

Version Control and Image Creation: Git fundamentals for Docker, preparing packaging for Git, Committing changes to an image.

Docker file Build Automation: Writing your first Docker file, Instructions: FROM, RUN, COPY, ADD, CMD, ENTRYPOINT.

Optimizing Images: Image layers and relationships, Union File Systems (UFS) and copy-on-write, Versioning images with pragmatic tags

Experiments:

1. **Creating and Versioning Docker Images using Git**
   Initialize a Git repository for a sample application project and track all source code changes using Git commands. Create a Docker image from the application, commit modifications to new image versions, and apply appropriate tagging strategies to maintain image versions systematically.
2. **Writing Dockerfiles for Application Deployment**
   Write a Dockerfile to containerize a simple web-based application. Use Dockerfile instructions such as FROM, RUN, COPY, ADD, CMD, and ENTRYPOINT to automate image creation. Build the image, run the container, and verify the successful deployment of the application in the browser or terminal.
3. **Optimizing Docker Images and Layer Management**
   Analyze Docker image layers generated during image creation and study the impact of instruction ordering on image size and cache usage. Modify the Dockerfile to minimize unnecessary layers and optimize build efficiency using lightweight base images and proper caching mechanisms.

UNIT III: Persistent Storage and Docker Networking (6 hrs)

Persistent Storage: Introduction to Volumes, Volume types: Bind mounts and Docker-managed volumes, Sharing data between containers and host, Data-packed volume containers.

Docker Networking: Protocols, Interfaces, and Ports, Network archetypes: Closed, Bridged, Joined, and Open, Container linking and service discovery, Exposing ports on the host.

Experiments:

1. **Working with Docker Volumes and Bind Mounts**

Create Docker-managed volumes and bind mounts to maintain persistent data storage across container restarts. Demonstrate data sharing between the host machine and containers, and between multiple containers. Observe the differences between bind mounts and Docker-managed volumes through practical examples.

2. **Implementing Inter-Container Communication**

Configure multiple Docker containers to communicate with each other using Docker networking. Create bridged and joined networks, enable container linking, and test service discovery mechanisms using container names and custom hostnames.

3. **Exposing Services and Port Mapping**

Deploy a web server container and expose its services to the host system using Docker port mapping techniques. Experiment with different network architectures, such as closed, bridged, and open. Verify accessibility of containerized services using browser and terminal-based tools

UNIT IV: Multi-Container Orchestration and Cluster Management (7 hrs)

Multi-Container Orchestration: Introduction to Docker Compose, Declarative environments with YAML, Scaling services up and down, Service dependencies and linking problems.

Cluster Management: Provisioning with Docker Machine, Introduction to Docker Swarm, Scheduling algorithms: Spread, BinPack, and Random, Service discovery in clusters.

Experiments:

1. **Developing Multi-Container Applications using Docker Compose**
   Create a multi-container application environment using Docker Compose YAML configuration files. Configure dependent services such as web server, database, and cache containers. Execute and manage all services collectively using Docker Compose commands.
2. **Scaling and Managing Services using Docker Compose**
   Implement service scaling for a containerized application using Docker Compose. Increase and decrease the number of running service instances dynamically and observe load balancing behavior. Analyze service dependency handling and container communication during scaling operations.
3. **Exploring Docker Swarm and Scheduling Algorithms**
   Initialize a Docker Swarm cluster and add worker nodes to the cluster environment. Deploy replicated services and study scheduling algorithms such as Spread, BinPack, and Random. Observe task distribution and service discovery mechanisms across cluster nodes.

UNIT V: CI/CD Pipelines and Distribution (6 hrs)

CI/CD Pipelines and Distribution: Automated builds on Docker Hub, Building Continuous Delivery pipelines with Jenkins and Mesos, Using Git Hooks for local automated builds. **Software Distribution:** Public vs Private registries, V2 Registry API, Securing registries with TLS and Authentication, Use Case: Consistent development environment for Node.js.

Experiments:

1. **Automating Docker Image Builds using Docker Hub**
   Create a Docker Hub repository and configure automated image builds from a GitHub source repository. Push source code changes to GitHub and verify automatic Docker image generation and version updates through Docker Hub integration.
2. **Building a Simple CI/CD Pipeline using Jenkins**
   Install and configure Jenkins for automating application build and deployment tasks. Create a Jenkins pipeline to fetch source code from GitHub, build a Docker image, execute automated build steps, and deploy the containerized application successfully.
3. **Configuring Private Docker Registries and Webhooks**
   Set up a private Docker registry with authentication and security configurations. Push and pull Docker images securely between local systems and the registry. Integrate webhook notifications to automate deployment or update activities whenever new images are uploaded.

Text Book(s): .

[1]. Sébastien Goasguen, "Docker Cookbook: Solutions and Examples for Building Distributed Applications", 1st Edition, O'Reilly Media, 2016.
[2]. Jeff Nickoloff, "Docker in Action", 1st Edition, Manning Publications, 2016.

REFERENCE BOOKS:

1. Jon Loeliger and Matthew McCullough, "Version Control with Git", 2nd Edition, O'Reilly Media (Recommended in T1 Preface).
2. James Murty, "Programming Amazon Web Services", O'Reilly Media (Recommended in T1 for AWS integration).

3. Bill Lubanovic, "Introducing Python", O'Reilly Media (Recommended for scripting in T1).

E-Resources and Other Digital Material:

[1]. Docker Inc. Official Docker Documentation [Internet]. Docker Documentation; [cited 2026 May 12]. Available from: https://docs.docker.com/
[2]. Goasguen S. Docker Cookbook GitHub Repository [Internet]. GitHub; [cited 2026 May 12]. Available from: https://github.com/how2dock/docbook.git
[3]. Nickoloff J. Docker in Action GitHub Repository [Internet]. GitHub; [cited 2026 May 12]. Available from: https://github.com/dockerinaction
[4]. GitHub Inc. Interactive GitHub Tutorials [Internet]. GitHub Training; [cited 2026 May 12]. Available from: http://training.github.com/
[5]. Docker Inc. Docker Registry API Specification [Internet]. Docker Documentation; [cited 2026 May 12]. Available from: https://docs.docker.com/registry/spec/api/

ETHICAL HACKING

Course Code	24IT382B		
Course Category:	Skill Course	**Credits:**	01
Course Type:	Practical	**Lecture - Tutorial - Practice:**	0-0-2
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	100 - 100

COURSE DESCRIPTION

This course focuses on the concepts of ethical hacking, cybersecurity principles, and techniques used to identify and mitigate vulnerabilities in computer systems and networks. It covers various phases of penetration testing, including reconnaissance, scanning, gaining access, maintaining access, and covering tracks. In addition to that, it also imparts knowledge of common attack vectors, system vulnerabilities, network security threats, and countermeasures. The objective of this course is to enable the student to analyse security risks, perform ethical hacking practices responsibly, and design secure systems to protect organizational assets from cyber threats.

COURSE OBJECTIVES

1. Understand the fundamentals of ethical hacking, cybersecurity principles, and the role of ethical hackers in protecting systems and networks.
2. Analyze various types of cyber threats, attack vectors, and vulnerabilities in computer systems and network infrastructures.
3. Develop skills to perform penetration testing activities including reconnaissance, scanning, and vulnerability assessment.
4. Apply ethical hacking tools and techniques to identify, exploit, and mitigate security weaknesses in a controlled environment.
5. Evaluate system security and implement appropriate countermeasures and best practices to safeguard organizational assets.

COURSE OUTCOMES

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

CO1: Summarize the fundamentals of ethical hacking, cybersecurity concepts, and various types of cyber threats. [K2]

CO2: Apply reconnaissance, scanning, and vulnerability assessment techniques to identify system and network weaknesses. [K3]

CO3: Demonstrate the use of ethical hacking tools and techniques to perform penetration testing in controlled environments. [K3]

CO4: Analyze different attack methods, security breaches, and their impact on systems and networks. [K4]

CO5: Design and implement appropriate security measures and countermeasures to protect systems from cyber attacks. [K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Introduction To Hacking (6 hrs)

Introduction To Hacking: Introduction to Hacking, Important Terminologies, What is a Penetration Test, Vulnerability Assessments versus Penetration Test, Penetration testing methodologies: OSSTMM, NIST, OWASP, Categories of Penetration Test, Types of Penetration Tests.

Experiments:

1. Perform network scanning and vulnerability identification using Nmap and OpenVAS.
2. Identify common vulnerabilities including SQL Injection, Cross-Site Scripting (XSS), and weak authentication using Burp Suite.
3. Conduct a mini penetration test on a simulated network with NIST and ISECOM OSSTMM.

UNIT II Network Sniffing (7 hrs)

Network Sniffing: Introduction, Types of sniffing, Hubs versus Switches, Promiscuous versus Non-Promiscuous mode, ARP Protocol Basics, ARP attacks, MAC Flooding, ARP Poisoning, Denial of service attacks, Tools of the Trade: Dsniff.

Experiments:

1. Install and configure Wireshark in promiscuous mode and capture live network packets and identify protocols such as ARP, TCP, UDP, HTTP, and DNS.
2. Perform ARP poisoning attack using tools such as arpspoof from the Dsniff package.
3. Simulate a switched network environment using Packet Tracer.

UNIT III: Footprints and Social Engineering (6 hrs)

Footprints and Social Engineering: Using Web tools for Foot printing: SamSpade, Web data Extractor, Conducting Competitive Intelligence, Using Domain Name System Transfers.

Port Scanning: Introduction to Port Scanning, Types of Port Scans, Using Port scanning tools: Nmap, Unicornscan, Conducting Ping Sweeps.

Experiments:

1. Select a public website/domain and perform footprinting using SamSpade and Web Data Extractor tools.

2. Configure a small lab network or virtual machines and perform ping sweeps to identify active hosts.
3. Use Nmap and Unicornscan to conduct TCP, SYN, and UDP port scans on target systems in the lab environment.

UNIT IV: Desktop and OS Vulnerabilities (6 hrs)

Desktop and OS Vulnerabilities: Windows OS Vulnerabilities, Tools for identifying vulnerabilities in Windows: Scanning Windows Using Nessus Essentials, Linux OS Vulnerabilities.

Hacking Web Servers: Understanding Web Applications, Understanding Web Application Vulnerabilities, Application Vulnerabilities and Countermeasures, Tools for Web attackers and Security testers, Web tools.

Experiments:

1. Install and configure Nessus Essentials. Perform a vulnerability scan on a Windows system and identify missing patches, weak configurations, and open ports.
2. Use Linux security assessment tools such as Nmap and Nikto to identify vulnerabilities in a Linux-based server environment.
3. Use web security testing tools like Burp Suite to identify vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and weak authentication.

UNIT V: Information Gathering Techniques (6 hrs)

Information Gathering Techniques: Active Information Gathering, Passive Information Gathering, Sources of Information Gathering, Copying Websites Locally,Yougetsignal.com, Tracing the Location, Trace route, ICMP Trace route, TCP Trace route, Usage, UDP Trace route, NeoTrace, Cheops-ng, Net craft, Google Hacking, Some basic parameters, TIP regarding file type, Google Hacking Database

Experiments:

1. Perform passive information gathering using Whois, Netcraft, and Yougetsignal.com.
2. Collect details including IP address, hosting provider, DNS records, server location, and technology stack.

Perform active reconnaissance using ping, traceroute, and TCP/UDP traceroute commands to map the network path to the target. Prepare a comparative report on passive vs active information gathering results

Text Book(s):

[1]. Rafay Baloch, “Ethical Hacking and Penetration Testing Guide”, CRC Press, 2015.
[2]. Robert S Wilson, Michael T. Simpson, Nicholas Antill, “Hands -On Ethical Hacking and Network Defense”, Fourth Edition, Cengage Learning, 2022

Reference Books:

[1]. Steven DeFino, Barry Kaufman, Nick Valenteen, “Official Certified Ethical Hacker Review Guide”, Cengage Learning, 2009.

[2]. Patrick Engebretson, “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy”, Syngress Basics Series – Elsevier, 2011.
[3]. Whitaker & Newman, “Penetration Testing and Network Defense”, CiscoPress, Indianapolis, 2006

E-resources and other digital material

[1]. Prof. Indranil Sengupta, Department of Computer Science & Engineering, IIT Kharagpur, NPTEL, Lecture Series on Ethical Hacking, 2024.
https://onlinecourses.nptel.ac.in/noc24_cs94/preview
[2]. NPTEL, IIT Kharagpur, Ethical Hacking Lecture Videos (YouTube Series), 2022.
https://www.youtube.com/watch?v=LQUocnZlBgQ
[3]. NPTEL, IIT Kanpur, Practical Cyber Security for Cyber Practitioners (including ethical hackingconcepts),2024.
https://www.youtube.com/watch?v=dm9xZIzDhwM
[4] Fatima Asif et al., Ethical Hacking and its Role in Cybersecurity, arXiv, 2024.
https://arxiv.org/abs/2408.16033

DOT NET PROGRAMMING

Course Code:	24IT382C		
Course Category:	Skill Course	**Credits:**	01
Course Type:	Theory	**Lecture - Tutorial - Practice:**	0-0-2
Prerequisites:	Programming Fundamentals	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	100 - 100

COURSE DESCRIPTION

This course provides a comprehensive introduction to C# programming and the .NET framework for developing modern software applications. It covers object-oriented programming concepts, type creation, inheritance, delegates, events, exception handling, and lambda expressions in C#. The course introduces Entity Framework Core for database connectivity, data modeling, querying, and manipulation using relational databases. Students also learn web application development using ASP.NET Core Razor Pages, HTTP concepts, dependency injection, and integration of Entity Framework Core with web applications. The course enables students to design, develop, and deploy data-driven and web-based applications using modern .NET technologies

COURSE OBJECTIVES

1. Understand the fundamentals of C# programming and the architecture of the .NET framework.
2. Develop object-oriented applications using classes, inheritance, delegates, events, lambda expressions, and exception handling techniques.
3. Apply Entity Framework Core for database connectivity, data modeling, querying, and manipulation in modern applications.
4. Analyze web development concepts and build dynamic web applications using ASP.NET Core Razor Pages.
5. Design and develop data-driven software applications integrating C#, Entity Framework Core, and ASP.NET Core technologies.

COURSE OUTCOMES

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

CO1 Summarize the Microsoft .NET Framework Architecture and its features such as delegates and Events.[K1]

CO2 Apply the object oriented features of Dot Net frame work in solving Realworld applications. [K2]

CO3 Understand how lambda expressions enhance code readability and reduce program complexity.[K3]

CO4 Implement modern database interactivity using the Entity framework for database connectivity.[K3]

CO5 Develop a dynamic web application using ASP.net core Razor pages.[K3]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	2		1		2								
CO2	3					2							2
CO3	3					2							2
CO4	3			2	3								2
CO5			3		3								3

COURSE CONTENT

UNIT I: Introduction to DotNet (6 hrs)

Introducing C# and .NET: Object Orientation, Type Safety, Memory Management, Platform Support, CLRs, BCLs, and Runtimes, Common Language Runtime, Base Class Library

Creating Types in C#: Classes, Fields, Constants, Methods, Instance Constructors, de-constructors, Object Initializers, this Reference, Properties, Indexers, Static Constructors, Static Classes, Finalizers.

Experiments:

1. Create a class to store and display student details using objects.
2. Write a program to demonstrate default and parameterized constructors.
3. Develop a class with multiple constructors to illustrate constructor overloading
4. Implement the concept of Indexers and identify the differences between Properties and Indexers

UNIT II: Advanced C# (6 hrs)

Inheritance: Polymorphism, Casting and Reference Conversions, Virtual Function Members, Abstract Classes and Abstract Members

Delegates: Writing Plug-In Methods with Delegates, Instance and Static Method Targets, Multicast Delegates, Built in Generic Delegates, Delegates Versus Interfaces.

Experiments:

1. Create an abstract class Shape with abstract methods to calculate area and perimeter.
2. Develop a program demonstrating single inheritance using Person and Student classes.
3. Implement multilevel inheritance using employee management classes.
4. Write a C# program that demonstrates the basic usage of delegates and multicaste delegate

UNIT III: Events & Expressions (6 hrs)

Events: Standard Event pattern, Event Accessors, Event Modifiers.

Lambda Expressions: Explicitly Specifying Lambda Parameter and Return Types. Anonymous methods. Try Statements and Exceptions, The catch Clause, The finally Block, Throwing Exceptions, Key properties of System. Exception, Common Exception types.

Experiments:

1. Create a delegate to perform arithmetic operations and invoke different methods using the same delegate.
2. Develop a calculator application using multicast delegates for multiple operations.
3. Create a program demonstrating the relationship between delegates and events.
4. Implement a simple event that is raised when a button is clicked or a task is completed.
5. Implement the concept for Exception Handling.

UNIT IV: Working with data using Entity Framework core (7 hrs)

Understanding modern databases: Understanding legacy Entity Framework, Understanding Entity Framework Core, Creating a console app for working with EF Core, Using a sample relational database, Using Microsoft SQL Server for Windows, Creating the Northwind sample database for SQL Server.

Setting up EF Core: Choosing an EF Core database provider, connecting to a database, defining the Northwind database context class

Defining EF Core models: Using EF Core conventions to define the model, Using EF Core annotation attributes to define the model, Using the EF Core Fluent API to define the model.

Querying EF Core models: Filtering included entities, Filtering and sorting products. Loading patterns with EF Core: Eager loading entities, Enabling lazy loading, Explicit loading entities. Manipulating data with EF Core: Inserting entities, Updating entities, Deleting entities, Pooling database contexts.

Experiments:

1. Create a database connection using Entity Framework Core and SQL Server.
2. Implement CRUD operations using Entity Framework Core.
3. Develop a student management system using Entity Framework Core.

UNIT V: Building Websites Using ASP.NET Core Razor Pages (8 hrs)

Building Websites Using ASP.NET Core Razor Pages

Understanding web development: Understanding HTTP, Using Google Chrome to make HTTP requests, Understanding client-side web development technologies. Understanding ASP.NET Core: Classic ASP.NET versus modern ASP.NET Core, Creating an empty ASP.NET Core project,

Exploring ASP.NET Core Razor Pages: Enabling Razor Pages, Adding code to a Razor Page. Using Entity Framework Core with ASP.NET Core: Configure Entity Framework Core as a service, Manipulating data using Razor Pages, Injecting a dependency service into a Razor Page

Experiments:

1. Develop Razor Pages with form handling and data validation.
2. Create CRUD operations using Razor Pages and Entity Framework Core.
3. Design and develop end-to-end web applications using ASP.NET Core Razor Pages and Entity Framework Core.

Text Book(s):

[1]. Mark J.Price, " C# 10 and .NET 6 – Modern Cross-Platform Development - Sixth Edition", Oreilly publications.Nov2021
[2].Joseph Albahari,"C#10 in a Nut Shell", Oreillypublications.Nov 2021

Reference Books:

[1]. Kemal Birer," ASP.NET Core for Jobseekers"bpb publications 2021.
[2]. Kogent Learning Solutions, "ASP.NET4.5 PROGRAMMING" Black Book, dreamtech press, 2013.
[3]. Scott Hanselman, Maira Wenzel, Modern Web Development with .NET 6 Ep1: Create a web UI with ASP.NET Core, https://docs.microsoft.com/en-us/shows/learn-live/modern-web-development-net6-ep01-create-web-ui-aspnet-core (16-05-2022)
[4]. RehanSaeed, Upgrading ASP.NET Core to .NET 6 & C# 10, https://techcommunity.microsoft.com/t5/web-development/upgrading-asp-net-core-to-net-6-amp-c-10/m-p/2927530 and https://www.youtube.com/watch?v=T6iP7QPWmPI (Microsoft) (16-05-2022)
[5]. Cam Sopar, Getting Started with Entity Framework Core, https://docs.microsoft.com/en-us/shows/entity-framework-core-101/getting-started-with-entity-framework-core (16-05-2022)

|.

LINUX PROGRAMMING

Course Code	24IT382D		
Course Category:	Program Core	**Credits:**	1
Course Type:	Theory	**Lecture - Tutorial - Practice:**	0-0-2
Prerequisites:	-	**Continuous Evaluation:** **Semester end Evaluation:** **Total Marks:**	100 100

COURSE DESCRIPTION

This course introduces Linux/Unix system programming with emphasis on shell scripting, system calls, file handling, and process management. It covers memory management, signals, and inter-process communication mechanisms such as pipes, shared memory, and semaphores. The course enables students to develop efficient system-level applications in a Linux environment.

COURSE OBJECTIVES

- To Understand fundamental knowledge of Linux/Unix operating system and shell programming.
- To understand file structures, system calls, and standard I/O operations in Linux.
- To develop skills in system-level programming for file and directory management.
- To understand process creation, memory management, and synchronization techniques.
- To enable students to design and implement inter-process communication using pipes, shared memory, and semaphores.

COURSE OUTCOMES

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

CO1: Summarize the fundamentals of Linux/Unix environment and demonstrate basic shell scripting. [K2]
CO2: Apply system calls and standard I/O functions for file and directory management. [K3]
CO3: Analyze process creation, control mechanisms, and signal handling in Linux systems. [K4]
CO4: Evaluate memory management and synchronization techniques for efficient resource utilization. [K5]
CO5: Design and implement inter-process communication mechanisms using pipes, shared memory, and semaphores. [K6]

MAPPING OF COURSE OUTCOMES TO PROGRAM OUTCOMES:

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

COURSE CONTENT

UNIT I: Shell Programming (6 hrs)

Getting Started: An Introduction to Unix, Linux and GNU, Programming Linux.

Shell Programming: A bit Philosophy, what is a shell, commonly used Linux shells, Pipes and Redirection, Shell as a programming Language, Shell Syntax.

Experiments:

1. Write and execute Linux commands to perform the following tasks:
    - Create a directory named Linux Lab
    - Create three text files inside it
    - Display the contents of the files using cat
    - Copy one file to another name
    - List all files with detailed permissions using ls -l
2. Create a shell script that demonstrates the use of:
    - Input redirection (<)
    - Output redirection (>)
    - Append redirection (>>)
    - Pipes (|)
3. Write a shell script to accept two numbers from the user perform addition, subtraction, multiplication, and division, display the results using proper formatting use conditional statements to check division by zero.

UNIT II : **Working with Files (8 hrs)**

Working with Files: Linux file structures- Directories, Files and Devices, System calls and Device drivers, Library functions, Low-Level File Access- write, read, open, Initial Permissions, Other System Calls for Managing Files, the standard I/O Library - fopen, fread, fwrite, fclose, fflush, fseek, fgetc, getc, and getchar, fputc, putc, and putchar, fgets and gets. Formatted Input and Output - printf, fprintf, and sprint, scanf, fscanf, and sscanf, Other Stream Functions, Stream Errors, Streams and File Descriptors.

Files and Directory Maintenance: chmod, chown, unlink, link, and symlink, mkdir and rmdir, chdir,getcwd and GREP.

Experiments:

1. Write a program using Linux system calls to:
    - Create a file using open()
    - Write data into the file using write()
    - Read the contents using read()
    - Display the contents on the screen
    - Close the file using close()
2. Write a program to perform student record management using standard I/O functions:
    - Accept student details from the user
    - Store the details in a file using fprintf() or fwrite()
    - Read and display the records using fscanf() or fread()
    - Use fseek() to access a particular student record from the file.
3. Perform the following Linux operations and verify the results:
    - Create and remove directories using mkdir and rmdir
    - Change file permissions using chmod
    - Change file ownership using chown
    - Create hard links and symbolic links using link and symlink
    - Display the current working directory using getcwd() and change it using chdir() in a C program.
4. Write a shell script to search for a pattern in a file using grep.
5. Write a shell script to extract email IDs from a text file using grep.

UNIT III: Scanning Directories (6 hrs)

Scanning Directories: Scanning Directories and Errors.

Linux environment: Program Arguments, Environment variables, Time and Date, Temporary Files, User information, Host information, Logging, Resources and Limits.

Experiments:

1. Write a shell script to scan a given directory and display the names of all files, subdirectories, and hidden files. Handle errors if the directory does not exist.
2. Write a shell script to demonstrate the use of command-line arguments and environment variables by displaying the username, home directory, shell path, and all arguments passed to the script.
3. Write a shell script to display current date and time, create a temporary file, show host information and logged-in user details, and append the information into a log file.

UNIT IV: Data Management (6 hrs)

Data Management: Managing Memory - Simple Memory Allocation, Allocating Lots of Memory, Abusing Memory, The Null Pointer, Freeing Memory, Other Memory Allocation Functions, File Locking- Creating Lock Files, Locking Regions, Use of read and write with Locking, Competing Locks, Other Lock Commands, Deadlocks.

Processes and Signaling: What is a process, Process structure, Starting new process, Signals

Experiments:

1. Write a program to demonstrate dynamic memory allocation using *malloc(), calloc(), realloc()*, and *free()* for storing and managing integer arrays.
2. Write a program to implement file locking using *fcntl()* and demonstrate read/write locking between multiple processes accessing the same file.
3. Write a program to create parent and child processes using *fork()* and demonstrate signal handling using *signal()* and *kill()* system calls.

UNIT V:

Inter-Process Communication: Pipes (6 hrs)

What Is a Pipe? Process Pipes, Sending Output to popen - Passing More Data, how popen Is Implemented. The Pipe Call, Parent and Child Processes - Reading Closed Pipes, Pipes Used as Standard Input and Output, Named Pipes: FIFOs - Accessing a FIFO, Advanced Topic: Client/Server Using FIFOs.

Shared Memory - shmget, shmat, shmdt, shmctl.

Semaphores - Semaphores, Semaphore Definition, A Theoretical Example, Linux Semaphore Facilities, Using Semaphores.

Experiments:

1. Write a program using pipe() to establish communication between parent and child processes. The parent process should send data through the pipe, and the child process should read and display the received message.
2. Write a program using shared memory system calls (shmget(), shmat(), shmdt(), and shmctl()) to share data between two processes.
3. Write a program using semaphores to synchronize multiple processes and control access to a shared file or shared memory segment.

Text Book(s):

[1]. Neil Matthew and Richard Stones “Beginning Linux Programming” 4th edition Wrox Publication.

[2] Unix and shell Programming Behrouz A. Forouzan, Richard F. Gilberg. Thomson

Reference Books:

[1]. Unix and Shell Programming, B. A. Forouzan and R. F. Gilberg, Cengage Learning.

[2]. Linux System Programming, OReilly, SPD.

[3].W. Richard. Stevens,” Advanced Programming in the UNIX Environment”, Pearson Education, 3rd edition, New Delhi, India.

E-resources and other digital material

[1]. Prof. Anand Iyer, Department of Computer Science & Engineering, IIT Madras, NPTEL, Lecture Series on Linux Programming & Scripting, 2024.

https://nptel.ac.in/courses/117106113
[2]. Prof. Chrysanthos Dellarocas, Sloan School of Management, MIT, The Linux Tutorial, 2025.
https://ocw.mit.edu/courses/15-564-information-technology-i-spring-2003/pages/lecture-notes/
[3]. Prof. Daniel Walkes, Department of Computer and Energy Engineering, University of Colorado Boulder, Linux System Programming and Introduction to Buildroot, 2023. https://www.coursera.org/learn/linux-system-programming-introduction-to-buildroot