SIPNA COLLEGE OF ENGINEERING & TECHNOLOGY, AMRAVATI

An Autonomous Institute Affiliated to

Sant Gadge Baba Amravati University, Amravati, Maharashtra (India)

(Approved by AICTE, New Delhi and Recognized by DTE, Maharashtra)

(Accredited With 'A+' Grade by NAAC)

Syllabus: Semester V and VI

Department of Information Technology

B.Tech. Information Technology with Multidisciplinary Minor

(Semester Pattern)

Effective from Academic Year 2026-27

Prepared By: Boards of Studies-Information Technology

Approved By: Academic Council -Sipna COET, Amravati

			30/3/2026	1.00
Chairman Board of Studies	Dean Academics	Chairman Academic Council	Date of Release	Version

Program:	B.Tech.(Information Technology)			Semester:			V	
Course:	Compiler Design			Code:			BTITPC14IT5T	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignment/Quizzes, Attendance, Viva-VOCE

Course Objectives:

1. Understand the structure and functioning of a compiler
2. Analyze lexical analysis and parsing techniques such as top-down and bottom-up parsing.
3. Comprehend syntax-directed translation techniques and the use of attributes in semantic analysis
4. Design intermediate code representations such as syntax trees, DAGs, and three-address codes
5. Understand run-time environments, memory organization, and activation records
6. Explore code generation and optimization strategies to improve the efficiency of generated code

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Explain the phases of compiler design and the role of each phase in program translation.	L2
CO-2	Solve the First & Follow and construct LL (1) Parsing Table	L3
CO-3	Construct parsers using bottom-up (LR) parsing methods such as SLR, CLR, LALR	L4
CO-4	Develop syntax-directed translation schemes and generate syntax trees.	L6
CO-5	Generate intermediate code representations and perform type checking and type conversion.	L5
CO-6	Understand the concept code generation and code optimization	L2

Unit I: Introduction to Compiler **(6 Hrs.)**

Introduction to Compiling: The structure of a compiler, Lexical Analysis: The role of lexical analyzer, input buffering, specification of tokens, recognition of tokens, The lexical analyzer generator Lex

Unit II: Syntax Analysis **(6 Hrs.)**

Syntax Analysis: The role of the parser, Review of context free grammar for syntax analysis: Parse Tree and Derivation, Ambiguity in Grammar, Elimination of left recursion and left factoring. Top-down parsing: recursive descent parsing, FIRST and FOLLOW, LL (1) grammars, Construction of predictive parsing tables

Unit III: Bottom-up parsing **(6 Hrs.)**

Bottom-up parsing: Handle pruning, Stack implementation of Shift Reduce Parsing, conflicts during shift reduce parsing Introduction to LR parsing: Simple LR, Items and the LR (0) Automation, The LR-Parsing algorithm, Construction of SLR parsing table, More powerful LR Parsers: canonical LR (1) Items, Constructing LR (1) sets of items and canonical LR (1) parsing tables, Constructing LALR parsing tables,

Unit IV: Syntax Directed Translation **(6 Hrs.)**

Syntax Directed Translation: Syntax directed definitions, Inherited and synthesized attributes, Evaluation orders of SDD's, Dependency Graphs, s-attributed definitions, L-attributed definition, Application of Syntax-Directed Translation, construction of syntax trees.

Unit V: Intermediate-Code Generation **(6 Hrs.)**

Intermediate-Code Generation: Variants of Syntax Trees: Directed Acyclic Graphs (DAG), Three Address Code, Type Checking, rules of type checking, type conversion.

Run Time Environments: Storage Organization, Static versus Dynamic Storage Organization, Stack Allocation of Space: Activation trees, Activation Records, Calling Sequences

Unit VI: Code Generation **(6 Hrs.)**

Code Generation: Issues in Design of a Code generator, The Target Language, Basic blocks and flow graphs. Optimization of Basic Blocks, Peephole Optimization and The Principal sources of Optimization.

Total Lecture 36 Hours

Textbooks:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman Compilers: "Principles, Techniques and Tools", Pearson Education Second Edition.

Reference Books:

1. D. M. Dhamdhere, Compiler Construction—Principles and Practice, (2/e), Macmillan India
2. Andrew Appel, Modern Compiler Implementation in C, Cambridge University press
3 K C. Louden "Compiler Construction—Principles and Practice" India Edition, CENGAGE
4 Bennett J.P., "Introduction to Compiling Techniques", 2/e (TMH).

MOOCs Links and additional reading, learning, video material

1. https://onlinecourses.nptel.ac.in/noc21_cs07/preview
2. https://www.udemy.com/course/compiler-design-rmm/?couponCode=PMNVD2025

Approved in..... Academic Council Meeting
Dated:- 30/3/2026

Program:	B.Tech.(Information Technology)			Semester:	V	
Course:	Compiler Design Lab			Code:	BTITPC15IT5P	
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

1. Understand the phases of a compiler and their interdependencies through hands-on programming
2. Implement lexical analyzers to recognize tokens from source code.
3. Develop parsers to construct parse trees or syntax trees.
4. Apply syntax-directed translation techniques to generate intermediate representations
5. Generate target code from intermediate code
6. Simulate code optimization techniques and understand their effect on performance

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Identify the fundamentals of compiler and its phases.	L1
CO-2	Use the powerful compiler generation tools such as Lex and Yacc.	L3
CO-3	Develop program for solving LL(1) parser problem.	L5
CO-4	Develop program for solving Shift reduce parser problems.	L5
CO-5	Design a program for the implementation of code generation.	L6
CO-6	Examine the various optimization techniques.	L4

General Guidelines: The provided sample list of programs is intended as a guide, and the subject teacher has the flexibility to create a customized set of experiments aligned with the Python programming curriculum.

Sr. No.	List of Practical
1	Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tab and new line.
2	Write a program to identify whether a given line is a comment or not

3	Design a lexical analyzer for given language and the lexical analyzer should list various tokens present in the input string.
4	Implement a Program to design Finite Automata and simulate the String which ends with '01'
5	Write a Program to implement Recursive Descent Parser
6	Write a Program to implement Shift Reduced Parser
7	Using JFLAP tool, create LL(1) parse table for a given grammar and hence Simulate LL(1) parsing
8	Using JFLAP tool, create SLR(1) parse table for a given grammar. Simulate parsing and output the parse tree in proper format.
9	Write a LEX Program for the implementation of lexical analyzer
10	Write a YACC Program for the implementation of calculator

Program:	B.Tech. (Information Technology)			Semester:			V	
Course:	Cloud Computing			Code:			BTITPC16IT5T	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Class Tests, Assignments, Quiz & Class Attendance

Course Objectives: To understand cloud computing concepts, architectures, service models, platforms, and security for building and managing cloud-based applications.

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

CO	Course Outcomes	BT Level
CO-1	Describe the basic structural units of a processor as well as hardware units of embedded systems.	L2
CO-2	Explain architecture of microcontroller, and processor-memory organization for embedded system.	L3
CO-3	Use knowledge of programming to do embedded programming in various languages and use of data structures for programming.	L1
CO-4	Examine the basic concepts of operating systems with real-time operating systems aspects.	L6
CO-5	Assess the Real-Time Operating System programming concepts with Design examples and case studies.	L4
CO-6	Design embedded systems based various applications using embedded software development process and tools.	L5

Unit I: Cloud Computing Fundamental, Architecture and Management **(6 Hrs)**

Computing Paradigm and various computing types, Cloud Computing Fundamentals: Motivation for Cloud Computing, The need for Cloud Computing, Defining Cloud Computing, Principles of Cloud Computing, Requirements of Cloud Services, Cloud Applications, Benefits and Drawbacks. Cloud Computing Architecture and Management: Introduction, Cloud Architecture, Network connectivity in Cloud Computing, Applications on the cloud, Managing Cloud, Migrating Application to cloud.

Unit II: Cloud Deployment and Service Models **(6 Hrs)**

Cloud Deployment Models: Introduction, Private Cloud, Public Cloud, Community Cloud, Hybrid Cloud. Cloud Service Models: Introduction, Infrastructure as a Service, Platform as a Service, Software as a Service, Other Cloud Service Models.

Unit III: Operating System and Virtualization **(6 Hrs)**

Types of Operating Systems, Role of OS in Cloud Computing, Features of Cloud OS. Application Environment: Need for Effective ADE, Application Development Methodologies, Cloud Application Development Platforms

and Cloud Computing API's. Virtualization: Introduction, Virtualization Opportunities, Approaches to Virtualization, Hypervisors, Virtualization to Cloud Computing.

Unit IV: Software Development in Cloud and Networking for Cloud Computing (6 Hrs)

Introduction, Different Perspectives on SaaS Development, New Challenges, Cloud-Aware Software Development Using PaaS Technology. Networking for Cloud Computing: Introduction, Overview of Data Center Environment, Networking Issues in Data Centers, Transport Layer Issues in DCNs.

Unit V: Cloud Service Providers (6 Hrs)

Introduction, EMC: IT, and captive cloud toolkit, Google: Platform, Storage, Cloud connect, Cloud Print and App Engine, Amazon Web Services: Elastic Compute Cloud, Simple storage, Simple Queue Service, Microsoft: Windows Azure, IBM Cloud models and IBM Smart Cloud, SAP Labs: SAP HANA Cloud Platform, Virtualization Services Salesforce: Sales Cloud and Service Cloud, Rackspace and VMware. Advances in Cloud Computing: Inter-cloud, Cloud Management, Mobile Cloud, Media Cloud, Interoperability and Standards, Cloud Governance Computational Intelligence in Cloud, Green Cloud, Cloud Analytics

Unit VI: Open-Source Support for Cloud and Security in Cloud Computing (6 Hrs)

Open-Source Support for Cloud: Introduction, Open-Source Tools for IaaS, Open-Source Tools for PaaS, Open-Source Tools for SaaS, Open-Source Tools for Research, Distributed Computing Tools for Management of Distributed Systems. Security in Cloud Computing: Introduction, Security Aspects: Data, Virtualization and Network Security, Platform-Related Security: Security issues in Cloud Service Models, SaaS, PaaS, IaaS security issues, Audit and Compliance: Disaster Recovery, Privacy and Integrity.

Total: 36 Hrs

Textbooks:

K. Chandrasekaran: Essentials of Cloud Computing, Edition, CRC Press Taylor & Francis Group.

Reference Books:

1. A. Shrinivasan, J. Suresh: Cloud computing a practical approach for learning and implementation, Pearson publication.

2. M.N. Rao: Cloud Computing, PHI Learning Pvt. Ltd, 2015.

3. Dr. Kumar Saurabh: Cloud computing, 2nd Edition, Wiley India 2012.

4. Rajkumar Buyya, James Broberg and Andrzej M. Goscinski: Cloud Computing: Principles and Paradigms, John Wiley & Sons, Inc. 2011.

5. Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, Cloud computing a practical approach, Tata McGraw-Hill, New Delhi – 2010.

6. Judith Hurwitz, Robin Bloor, Marcia Kaufman and Fern Halper, "Cloud computing for dummies" Wiley Publishing, Inc, 2010.

Program:	B.Tech. (Information Technology)			Semester:	V	
Course:	Cloud Computing Lab			Code:	BTITPC17IT5P	
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	0	2	1	30	20	50

Course Objectives: To enable students to design, develop, deploy, and secure cloud-based applications using cloud architecture, service models, virtualization, and open-source or public cloud platforms through a structured mini-project approach.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO1	Identify cloud computing requirements and design basic architecture for a given problem.	L2
CO2	Choose appropriate cloud service and deployment models to meet specific application needs.	L3
CO3	Configure and deploy virtual environments and cloud platforms for application hosting.	L4
CO4	Develop and deploy cloud-based applications using public or open-source cloud tools.	L6
CO5	Implement cloud security features including authentication, authorization, and data protection mechanisms.	L5
CO6	Test, document, and present cloud applications demonstrating integration of services and performance validation.	L3

Experiment

To design and implement a mini-project in a cloud environment by applying cloud computing fundamentals, architecture, service models, virtualization, platform services, and security practices covered in theory, using open-source tools or public cloud platforms.

While developing a cloud-based mini-project, students should follow these practical steps:

1. Understanding the Cloud Requirement and Architecture Design.
2. Select Suitable Cloud Service Model.
3. Virtualization and Environment Setup.
4. Cloud Application Development and Deployment.
5. Use of Open-Source Cloud Tools (as applicable).

6. Implement Security Mechanisms.
7. Cloud Service Integration and APIs.
8. Project Deployment, Testing, and Documentation.

Program:	B. Tech. (Information Technology)			Semester:			V	
Course:	Artificial Intelligence			Code:			BTITPC18IT5T	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignment/Quizzes, Attendance, Viva-Voce

Course Objectives:

- Introduce students to the foundational concepts, history, and evolution of Artificial Intelligence (AI) while discussing its risks and benefits in modern applications.
- Familiarize students with the concepts of intelligent agents, their structures, and environments, and distinguish between AI and Machine Learning (ML).
- Equip students with knowledge of various search strategies, including both uninformed and informed search techniques, and their application in problem-solving within AI.
- Teach students how to represent knowledge using Predicate Logic, and explore techniques for logical reasoning, unification, and question answering in AI systems.
- Provide an overview of different types of learning, focusing on the principles of supervised, unsupervised, and reinforcement learning, and introduce key machine learning algorithms.
- Develop skills in data preprocessing, feature selection, dimensionality reduction, and hyperparameter tuning to optimize machine learning models.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Explain the basic concepts, evolution, applications, and classification of AI, and differentiate between Artificial Intelligence and Machine Learning.	L2
CO-2	Analyze different types of intelligent agents, their rationality, and environmental characteristics, and evaluate the Turing Test as a measure of intelligence.	L3
CO-3	Formulate problem spaces using production systems and graphs, and apply basic search strategies to solve AI problems.	L3
CO-4	Implement uninformed and informed search strategies using appropriate data structures and evaluate their suitability for different AI problems.	L3
CO-5	Represent knowledge using predicate logic, perform resolution and unification, and design simple knowledge-based agents for reasoning and problem-solving.	L4
CO-6	Describe modern AI concepts, distinguish supervised and unsupervised learning, and apply basic neural network structures to simple AI problems.	L2

Unit I: Introduction to Artificial Intelligence **(6 Hrs.)**

Introduction: Foundation and History of AI, Overview of AI problems, Evolution of AI, Applications of AI, Classification/Types of AI. Artificial Intelligence vs Machine learning. Intelligent Agent: Types of AI Agent, Concept of Rationality, nature of environment, structure of agents. Turing Test in AI.

Unit II: Problem Solving **(6 Hrs.)**

Problems, Problem Space and Search: Production Systems, Problem Characteristics, Production System Characteristics, Issues in the Design of Search Programs. Problem trees and graphs. **Search Algorithms in Artificial Intelligence**: Terminologies, Properties of Search Algorithms, Types of Search Algorithms.

Unit III: Search Strategies **(6 Hrs.)**

Uninformed Search Strategies: Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search. Data Structures for BFS and DFS with a programmatic approach.

Unit IV: Informed Search Strategies **(6 Hrs.)**

Generate-and-Test, Hill Climbing, Best-first Search, A* Algorithm, Problem Reduction, AND-OR Graphs, The AO* Algorithm, Constraint Satisfaction, Minmax Algorithm. Means-Ends Analysis.

Unit V: Knowledge Representation using Predicate Logic **(6 Hrs.)**

Knowledge Representation and approaches, representing simple facts in logic, Resolution in Propositional Logic and Predicate Logic, Unification Algorithms, Question Answering, and Natural Deduction. Knowledge-Based Agent in Artificial Intelligence: Architecture, Approaches to designing a knowledge-based agent.

Unit VI: Modern AI **(6 Hrs.)**

Introduction to modern Artificial Intelligence, Supervised vs Unsupervised Learning, Role of data in modern AI systems, Types of data: Structured and Unstructured, Training and Testing datasets, Importance of data preprocessing. Concept of Artificial Neural Networks (ANN), Structure of neural networks: input, hidden, and output layers.

Total Lecture **36 Hours**

Textbooks:

1. Artificial Intelligence – Elaine Rich, Kevin Knight, Nair (Third Edition) [Mc Graw Hill]
2. Artificial Intelligence: A Modern Approach by Stuart Russell & Peter Norvig (Pearson -4th Ed.)
3. A Textbook of Machine Learning by SaiKat Dutt Subramanian Chandramouli and Amit Kumar Das, Pearson
4. Artificial Intelligence -Ela Kumar (Distributed By-Wiley Publication)

Approved in 4th Academic Council Meeting
Dated:- 30/3/2026

Reference Books:

1. Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd edition, Prentice Hall
2. Introduction to Artificial Intelligence & Expert Systems, Dan W Patterson, PH!., 2010 S Kaushik, Artificial Intelligence, Cengage Learning, 1st ed.2011
3. Luger, G.F. 2008. Artificial Intelligence -Structures and Strategies for Complex Problem Solving, 6th edition, Pearson.

MOOCs Links and additional reading, learning, video material

1. https://onlinecourses.nptel.ac.in/noc22_cs56/
2. NPTEL Course: Fundamentals of Artificial Intelligence https://elearn.nptel.ac.in/ shop /nptel/fundamentals-of-artificial- intelligenC~e1

Program: B.Tech. (IT)				Semester: V		
Course: Artificial Intelligence - Lab				Code: BTITPC19IT5P		
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

- Understand the fundamentals of Artificial Intelligence, intelligent agents, and their environments.
- Develop simple AI agents using rule-based, model-based, goal-based, and utility-based approaches.
- Apply search algorithms like BFS, DFS, Best-First, and A* to solve AI problems.
- Implement constraint satisfaction and optimization problems in AI.
- Apply supervised learning techniques with data preprocessing in Python for predictive modeling.
- Analyze and evaluate AI solutions using appropriate algorithms and heuristics for decision-making

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Recall and describe the basic concepts of Artificial Intelligence, intelligent agents, and AI tools.	L1
CO-2	Explain the functioning of Simple Reflex, Model-Based Reflex, Goal-Based, and Utility-Based agents in AI environments.	L2
CO-3	Implement BFS, DFS, and Best-First search algorithms to solve AI search problems.	L3
CO-4	Apply search with heuristic functions and solve constraint satisfaction problems in Python.	L4
CO-5	Develop supervised learning models (e.g., KNN, Decision Tree) with proper data preprocessing using Python.	L5
CO-6	Design and integrate AI agents and algorithms to solve complex real-world problems effectively.	L6

General Guidelines: The provided sample list of programs is intended as a guide, and the subject teacher can create a customized set of experiments aligned with the **Artificial Intelligence Lab** programming curriculum.

Practical. No.	List of Practical's
1	Introduction to Artificial Intelligence and different tools used for creating AI applications.
2	Create a Simple Reflex Agent in Python for a rule-based environment using Langflow.
3	Create a Model-Based Reflex Agent using internal state representation using Langflow
4	Create a Goal-Based Agent to solve a problem using search techniques using Langflow
5	Create a Utility-Based Agent for decision making using a utility function using Langflow.

6	Write a program to implement the breadth-first search algorithm.
7	Write a program to implement depth-first search algorithm.
8	Write a program to implement Best First Search algorithm.
9	Write a program to implement A* search algorithm using heuristic function in Python.
10	Write a program to Solve constraint satisfaction problem.
11	To implement a basic supervised learning model (e.g., KNN/Decision Tree) with data preprocessing using Python.

Program:	B. Tech. (Information Technology)			Semester:			V	
Course:	Data Science			Code:			BTITPE01IT5T	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Class Tests, Assignments, Quiz & Class Attendance

Course Objectives: Understand basics of data science and its statistics.

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

CO	Course Outcomes	BT Level
CO-1	Understand the basics of Data Science and its relation with machine learning	L4
CO-2	Apply data preprocessing methods on open access data and generate quality data for analysis	L4
CO-3	Apply and analyze classification and regression data analytical methods for real life Problems.	L3
CO-4	Apply different data analysis techniques to understand the data.	L3
CO-5	Analyze the data using suitable method; visualize using the open-source tool.	L4
CO-6	Model multi-dimensional data and visualize it using appropriate tool	L4

Unit I: Introduction to Data Science **(6 Hrs.)**

Defining data science and big data, Recognizing the different types of data, Gaining insight into the data science process, Data Science Process: Overview, Different steps, Machine Learning Definition and Relation with Data Science

Unit II: Statistics and Probability basics for data Analysis **(6 Hrs.)**

Statistics: Describing a Single Set of Data, Correlation, Simpson's Paradox, Some Other Correlational Caveats, Correlation and Causation Probability: Dependence and Independence, Conditional Probability, Bayes's Theorem, Random Variables, Continuous Distributions, The Normal Distribution, The Central Limit Theorem

Unit III: Data Analysis in depth **(6 Hrs.)**

Data Analysis Theory and Methods: Clustering –Overview, K-means- overview of method, determining number of clusters, Association Rules- Overview of method, Apriori algorithm, evaluation of association rules, Regression-Overview of linear regression method, model description. Classification- Overview, Naive Bayes classifier

Unit IV: Advanced Data Analysis **(6 Hrs.)**

Decision Trees: What Is a Decision Tree? Entropy, The Entropy of a Partition, Creating a Decision Tree, Random Forests Neural Networks: Perceptrons, Feed-Forward Neural Networks, Backpropagation, Example: Defeating a CAPTCHA MapReduce: Why MapReduce? Examples like word count and matrix multiplication

Unit V: Basics of Data Visualization **(6 Hrs.)**

Introduction to data visualization, challenges of data visualization, Definition of Dashboard, Their type, Evolution of dashboard, dashboard design and principles, display media for dashboard. Types of Data visualization: Basic charts scatter plots, Histogram, advanced visualization Techniques like streamline and statistical measures, Plots, Graphs, Networks, Hierarchies, Reports.

Unit VI: Data visualization of multidimensional data **(6 Hrs.)**

Need of data modeling, Multidimensional data models, Mapping of high dimensional data into suitable visualization method- Principal component analysis, clustering study of High dimensional data.

Total: 36 Hrs

Textbooks:

1. Data Mining: Concepts and Techniques, 3rd Edition. Jiawei Han, Micheline Kamber, Jian Pei.
2. Data Science from Scratch: Joel Grus, O'Reilly Media Inc., ISBN: 9781491901427 Information visualization perception for design, colin ware, MK publication

Reference Books:

1. Big data black book, Dream tech publication.
2. Getting Started with Business Analytics: Insightful Decision-Making, David Roi Hardoon, GalitShmueli, CRC Press
3. Business Analytics, James R Evans, Pearson
4. Python Data science Handbook, Jake VanderPlas, Orielly publication

MOOC Links:

1. https://nptel.ac.in/courses/106/106/106106179/
2. https://nptel.ac.in/courses/106/106/106106212/
3. https://nptel.ac.in/courses/106/105/106105174/

Program:		B. Tech. (Information Technology)		Semester:		V
Course:		Data Science Lab		Code:		BTITPE02IT5P
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives: To apply and analyze data through data visualization techniques.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO1	Understand data structures and statistics in R	L2
CO2	Understand and apply basics of ggplot.	L5
CO3	Implement regression analysis and interpret model results.	L5
CO4	Apply pre-processing/Data Cleaning Techniques.	L4
CO5	Apply Time series plotting and analysis	L5
CO6	Apply visualization Techniques.	L5

General Guidelines: Example: - Minimum 8 practicals are to be conducted covering entire syllabus.

Experiment List

1) Introduction and Installation of R Language/Visualization software.
2) Load built-in datasets (mtcars, iris, or a CSV file), explore its structure, and summarize key statistics.
3) Plot a scatter plot for mpg vs hp from the mtcars dataset and color by cyl
4) Perform a linear regression model on the mtcars dataset (mpg as dependent variable, hp as independent).
5) Create a sample dataset with missing values and demonstrate how to handle them (na.omit(), is.na()).
6) Use the AirPassengers dataset for basic time series visualization and decomposition.
7) Use Netflix Movies and TV Shows dataset from Kaggle and perform following operation : 1. Make a visualization showing the total number of movies watched by children 2. Make a visualization showing the total number of standup comedies 3. Make a visualization showing most watched shows. 4. Make a visualization showing highest rated show Make a dashboard (DASHBOARD A) containing all these above visualizations.
8) Use any data set from Kaggle and perform complete analysis.

Program:	B. Tech. (IT)			Semester:	V			
Course:	Computer Graphics and Image Processing			Code:	BTITPE03IT5T			
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignments, Class Tests, Viva Voce, Case Study Report, Attendance

Course Objectives: By the end of this course, students will be able to:

1. Understand the fundamentals and applications of computer graphics and image processing.
2. Learn and implement core algorithms for graphics rendering and geometric transformations in 2D and 3D.
3. Explore image representation techniques and perform basic image manipulation and enhancement.
4. Apply spatial and frequency domain techniques for image filtering and analysis.
5. Understand various image compression and segmentation techniques used in real-world applications.
6. Gain hands-on experience using tools such as OpenGL or OpenCV for practical implementation.

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

CO	Course Outcomes	BT Level
CO-1	Explain the basic principles and architecture of computer graphics systems and output devices.	L2
CO-2	Implement 2D and 3D geometric transformations, projections, and clipping algorithms using appropriate mathematical models.	L3
CO-3	Apply line, circle drawing algorithms and rendering techniques to generate graphical primitives.	L3
CO-4	Describe image fundamentals, color models, and perform basic point-based image processing operations.	L3
CO-5	Apply image enhancement, filtering, edge detection, and morphological operations in both spatial and frequency domains.	L3
CO-6	Analyze and implement image compression and segmentation techniques for practical applications such as face detection or OCR.	L4

Unit I: Introduction to Computer Graphics **(6Hrs.)**

Overview and applications of computer graphics, Graphics pipeline and architecture, Video display devices (CRT, LCD, LED, Plasma), Output primitives: Points, lines, circles, Line drawing algorithms: DDA, Bresenham's, Circle drawing: Midpoint and Bresenham's circle algorithm, Scan conversion and rasterization

Unit II: 2D Graphics and Transformations **(6Hrs.)**

2D coordinate systems, Geometric transformations: Translation, Rotation, Scaling, Shearing, Reflection, Homogeneous coordinates, Matrix representation of 2D transformations, 2D viewing: Window-to-viewport transformation, Clipping algorithms: Cohen-Sutherland, Liang-Barsky

Unit III: 3D Graphics and Transformations **(6Hrs.)**

3D coordinate systems and transformations, Projection techniques: Orthographic and perspective projections, 3D object representations: Polygonal meshes, surface modeling, Hidden surface removal: Z-buffer, Painter's algorithm, Basics of lighting and shading: Ambient, Diffuse, Specular, Introduction to OpenGL (or any relevant graphics library)

Unit IV: Introduction to Image Processing **(6Hrs.)**

Image fundamentals: Pixels, resolution, color models (RGB, CMYK, HSI, YCbCr), Image file formats: BMP, PNG, JPEG, TIFF, Image acquisition and digitization, Basic image operations: Arithmetic and logical operations, Point processing: Contrast stretching, Thresholding, Histogram equalization

Unit V: Image Enhancement and Filtering **(6 Hrs.)**

Spatial domain filters: Smoothing and sharpening filters, Frequency domain processing: Fourier Transform and applications, Noise models and noise reduction techniques, Edge detection: Sobel, Prewitt, Laplacian, Canny operators, Morphological operations: Erosion, Dilation, Opening, Closing

Unit VI: Image Compression, Segmentation and Applications **(6Hrs.)**

Image compression: Lossy vs. Lossless, JPEG, Run-Length Encoding, Huffman coding Image segmentation: Region growing, Watershed, K-means clustering, Color image processing basics, Applications: Face detection, OCR, Medical imaging, Industrial inspection

Total Lecture 36 Hours

Textbooks:

1. **"Computer Graphics: Principles and Practice"**, By James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes
2. **"Digital Image Processing"**. *By Rafael C. Gonzalez and Richard E. Woods*

Reference Books:

1. **"Fundamentals of Computer Graphics"**, *By Peter Shirley et al.*
2. **"Computer Graphics with OpenGL"**, *By Donald Hearn and M. Pauline Baker*
3. **"Image Processing, Analysis, and Machine Vision"**, *By Milan Sonka, Vaclav Hlavac, Roger Boyle*

Program:	B. Tech. (IT)			Semester:		V
Course:	Computer Graphics and Image Processing Lab			Code:		BTITPE04IT5P
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives: By the end of this lab course, students will be able to:

- Understand and implement basic graphics algorithms such as line, circle, and ellipse drawing.
- Perform 2D and 3D geometric transformations and clipping in graphical systems.
- Develop curve generation algorithms using Bezier and B-spline techniques.
- Gain practical experience in image processing tasks such as enhancement, filtering, and morphological operations.
- Apply image analysis techniques such as edge detection, segmentation, and compression using Python (OpenCV) or MATLAB.
- Strengthen programming skills and visualization abilities through graphical and image processing projects.

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

CO	Course Outcomes	BT Level
CO-1	Implement and compare fundamental line and circle drawing algorithms like DDA, Bresenham's, and Midpoint methods.	L3 & L4
CO-2	Apply 2D and 3D geometric transformations and viewing operations using matrix representations.	L3
CO-3	Develop algorithms for line clipping and curve generation using techniques such as Cohen-Sutherland and B-spline curves.	L6
CO-4	Perform basic image processing tasks including resizing, cropping, grayscale conversion, and histogram operations.	L3
CO-5	Apply spatial domain filtering and edge detection algorithms for image enhancement and analysis.	L3
CO-6	Implement morphological operations and perform basic image segmentation and compression techniques.	L3

General Guidelines: Minimum 8 Experiments are to be conducted covering entire syllabus

Expt. No.	List of Experiments
1	Line Drawing using DDA and Bresenham's Algorithm
2	Circle and Ellipse Drawing Algorithms
3	2D Geometric Transformations
4	Clipping Algorithms
5	Bezier and B-Spline Curves

6	3D Transformations and Simple Projection
7	Basic Image Operations: Read, display, resize, crop, and convert color images to grayscale.
8	Image Enhancement Techniques
9	Spatial Filtering
10	Edge Detection Techniques
11	Morphological Operations
12	Image Compression and Segmentation

Program:	B. Tech. (Information Technology)	Semester:	V
Course:	Advance JavaScript Programing with Framework	Code:	BTITPE05IT5T

Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	30	30	60	100

Methods of Teacher Assessment (TA): Continuous Assessment (40%), Assignment/Quizzes (40%), Attendance (20%)

Course Objectives:

- Understand the core concepts and syntax of JavaScript including ES6+ features.
- Apply JavaScript to build dynamic, interactive client-side web applications.
- Work with arrays, objects, DOM, and events to manipulate HTML content effectively.
- Handle asynchronous operations using callbacks, promises, and the Fetch API.
- Develop component-based frontend applications using ReactJS, including routing and state management.
- Gain foundational understanding of Angular concepts such as modules, services, data binding, and routing.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Write clean, modular JavaScript code using control structures, functions, and ES6+ syntax.	L2
CO-2	Manipulate the DOM and handle user-driven events using JavaScript and jQuery.	L3
CO-3	Handle asynchronous operations and integrate REST APIs using Fetch API and Promises.	L4
CO-4	Build single-page applications using ReactJS with component-based architecture, state, and routing.	L5
CO-5	Apply React Context API and Axios for state sharing and external data consumption.	L4
CO-6	Develop basic Angular applications using components, services, data binding, and routing.	L6

Unit I: Core JavaScript – Fundamentals & ES6+ **(6 Hrs.)**

Introduction to JavaScript, Script placement in HTML, JavaScript execution context, 'use strict' directive and its importance, Variables: var, let, const, Data types: string, number, boolean, null, undefined, symbol, Type conversion and typeof, Operators: arithmetic, assignment, comparison, logical, bitwise, ternary, Control structures: if, else if, switch, Loops: for, while, do-while, break, continue, Functions: declaration, expression,

arrow functions, return, default parameters, Scope and Hoisting: global, block, lexical, Closures and use cases, ES6+ features: template literals, spread and rest operators, destructuring

Unit II: JavaScript Arrays, Objects, DOM, Events and jQuery **(6 Hrs.)**

Arrays: creation, indexing, looping (for, forEach, for..of), Array methods: push, pop, splice, slice, map, filter, reduce, Objects: creation, properties, nested objects, loops, Object methods: keys, values, entries, DOM basics and element selection, Modifying content and attributes: innerText, innerHTML, setAttribute, Styling using style and classList, Event handling: click, change, input, addEventListener, event object, delegation, jQuery introduction, Selecting and modifying elements with jQuery, jQuery events and effects: click, hover, hide, show, slide, fade, Basic form handling using jQuery: reading input values, setting values, form validation

Unit III: Asynchronous JavaScript, AJAX and Fetch API **(6 Hrs.)**

Synchronous vs asynchronous behavior, Timers: setTimeout, setInterval, Callbacks and callback hell, Promises: then, catch, chaining, async and await syntax, XMLHttpRequest basics: GET and POST, reading response and status, Fetch API: GET and POST, handling JSON and errors, Understanding HTTP status codes: 200, 404, 500, JSON: stringify and parse, Consuming REST APIs using fetch, Rendering API data in HTML

Unit IV: ReactJS – Core Concepts, Hooks, and Forms **(6 Hrs.)**

Introduction to React and Virtual DOM, Setting up React using Create React App, JSX syntax and usage, Functional components and props, State management using useState, Event handling in React, Conditional rendering, List rendering and keys, React Hooks: useEffect, useRef, Form handling with controlled components, Form validation using Formik.

Unit V: ReactJS – Routing, Context, and API Integration **(6 Hrs.)**

React Router basics: BrowserRouter, Routes, Route, useNavigate, Route parameters, React Hook: useContext for global state sharing, API integration using fetch and axios, Handling loading and error states in API calls, Organizing folder structure: components, hooks, and API services

Unit VI: Angular – Fundamentals, Binding, Services & Routing **(6 Hrs.)**

Introduction to Angular, Angular CLI and project structure, TypeScript basics for Angular: types, interfaces, classes, Components and Modules: creation and structure, Data Binding: interpolation, property binding, event binding, two-way binding using ngModel, Built-in Directives: ngIf, ngFor with syntax and usage scenarios, Creating and using Services, Dependency Injection in Angular, Angular Routing: configuring routes with RouterModule, navigating between components, using route parameters, Overview of Angular Lifecycle Hooks (intro only), Consuming REST APIs using HttpClientModule, making GET and POST requests, displaying API data in components

Total Lecture 36 Hours

Textbooks:

1. **Flanagan, David.** ***JavaScript: The Definitive Guide*****, 7th Edition** *O'Reilly Media, 2020*

2. Chaffer, Jonathan and Swedberg, Karl. *Learning jQuery: A Hands-on Guide to Building Rich Interactive Web Frontends*, 4th Edition Packt Publishing
3. **Banker, Alex. *Learning React: Functional Web Development with React and Redux*, 3rd Edition** *O'Reilly Media, 2022*
4. **Chandermani and Banerjee, Shyam Seshadri. *Angular Up and Running: Learning Angular, Step by Step*, 2nd Edition**

MOOCs Links and additional reading, learning, video material

1. Coursera, https://www.coursera.org/learn/javascript-basics
2. Eloquent JavaScript (Free Book), MDN JavaScript Guide, https://eloquentjavascript.net/

Program:	B. Tech. (Information Technology)			Semester:	V	
Course:	Advance JavaScript Programming with Framework Lab			Code:	BTITPE06IT5P	
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

- To apply modern JavaScript (ES6+) features to build structured and interactive programs.
- To build responsive and dynamic web pages using DOM manipulation and jQuery for client-side interactivity.
- To develop asynchronous programs using Fetch API and integrate external APIs for data interaction.
- To design single-page applications using ReactJS, utilizing components, hooks, and routing.
- To integrate advanced React concepts like Context API and external APIs for real-time web applications.
- To construct modular Angular applications using services, routing, and component architecture.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Implement JavaScript programs using functions, control structures, arrays, and ES6+ syntax.	L3
CO-2	Manipulate HTML elements dynamically using JavaScript and jQuery to create responsive interfaces.	L3
CO-3	Handle asynchronous operations and REST API data using Promises and Fetch API.	L4
CO-4	Develop component-based React apps using state management and routing techniques.	L5
CO-5	Build scalable React applications with Context API and API-driven components.	L5
CO-6	Create Angular applications with data binding, routing, and service-based data flow.	L6

General Guidelines: The provided sample list of programs is intended as a guide, and the subject teacher has the flexibility to curate a customized set of practical's aligned with the curriculum.

Sr. No.	List of Practical
1	**Create a JavaScript calculator** that performs basic arithmetic using functions, if-else, and switch.

2	**Write a program** that demonstrates variable hoisting and scope using var, let, and const.
3	**Create arrow function variants** of a simple greeting and a sum function using ES6 syntax.
4	**Build a student marksheet app** using arrays and objects, allowing data entry and average calculation.
5	**Create a form validator** using JavaScript and add error messages dynamically via DOM manipulation.
6	**Build a to-do list** where users can add, delete, and mark items using DOM and event listeners.
7	**Enhance the to-do list using jQuery**: use jQuery selectors and animation to show/hide elements.
8	**Create a fake loading screen** using setTimeout and display "Data Loaded" after 3 seconds.
9	**Use the Fetch API** to display data (e.g., name, email) from https://jsonplaceholder.typicode.com/users.
10	**Handle Fetch error**: simulate a broken API URL and show custom error message on screen.
11	**Create a React app** that displays user information using props and functional components.
12	**Build a counter app** using useState to increase/decrease/reset a number.
13	**Create a clock app** that updates every second using useEffect.
14	**Build a contact form** using React, bind input fields, and show real-time form data on screen.
15	**Create a multi-page React app** using React Router with Home, About, and Contact pages.
16	**Build a user list component** using axios and display users from JSONPlaceholder.
17	**Add loading and error states** while fetching data from an API using useState and useEffect.
18	**Use React Context** to create a theme switcher (Light/Dark) that applies styles across components.
19	**Create a basic Angular app** with two components: ProductList and ProductDetails using property and event binding.
20	**Use Angular HttpClient** to fetch and display data from a public API and navigate between pages using RouterModule.

Select Any 2 Mini Project, Sample reference given (React + Angular)

Mini Project 1: ReactJS – Student Feedback Portal

Mini Project 2: Angular – Task Tracker App

Mini Project 3: ReactJS – Weather Dashboard

Mini Project 4: Angular – Online Product Catalog

Program:	B. Tech. (Information Technology)			Semester:	V	
Course:	Comm. Engg. Project/Field Project			Code:	BTITFP01IT5P	
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	50	-	50

Course Objectives:

To enable students to design and develop a real-world software or hardware-based solution by applying engineering principles, teamwork, and project management skills.

Course Outcomes:

CO	Course Outcomes	BT Level
CO1	Identify a real-world problem and define appropriate project objectives.	L4
CO2	Apply core computer science concepts to design an effective system solution.	L3
CO3	Conduct surveys and case studies to collect relevant data using appropriate tools and methodologies.	L3
CO4	Analyze the collected data using analytical and critical thinking skills to understand problem patterns and causes.	L4
CO5	Propose feasible and innovative solutions to address the identified community problems.	L6
CO6	Effectively present and communicate project findings and solutions to the community	L2

General Guidelines:

1) A group of 10 to 15 students shall select a project domain from the provided list. The selection is not restricted to the listed domains; students may propose an alternative domain with prior approval from the Chairman of the Board of Studies (BoS).
2) Students will prepare a case study and will deliver it in nearby community.
3) Model/Mobile app/Web app could also be prepared.

Domain List

1. Healthcare & Biomedical Applications
2. Agriculture & Smart Farming
3. Education & E-Learning Systems
4. Banking & Financial Applications
5. Retail & E-Commerce Systems
6. Transportation & Traffic Management
7. Smart City Applications
8. Energy & Power Management
9. Manufacturing & Production Systems
10. Environment & Pollution Monitoring
11. Security & Surveillance Systems
12. Social Welfare & E-Governance
13. Tourism & Hospitality Management
14. Logistics & Supply Chain Management
15. Sports & Fitness Applications
16. Library & Information Management
17. Automobile & Vehicle Management
18. Real Estate & Property Management
19. Media & Content Management
20. Human Resource & Office Automation

Program:	B.Tech.(IT)			Semester:	VI			
Course:	Design & Analysis of Algorithm			Code:	BTITPC20IT6T			
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE- II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA):

Course Objectives:

1. To understand the basic concept of algorithms and their performance analysis.
2. To introduce the divide and conquer design technique and its application in solving problems.
3. To develop the ability to apply greedy methods to optimization problems like MST and shortest paths.
4. To introduce dynamic programming and its use in solving complex optimization problems.
5. To introduce backtracking and graph traversal techniques for solving complex decision problems.
6. To introduce the concepts of computational complexity, including P, NP, and NP-Completeness.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Students will be able to define an algorithm and explain its performance parameters.	L1, L2
CO-2	Students will be able to apply the divide and conquer strategy to solve problems like searching and sorting.	L3
CO-3	Students will be able to implement greedy strategies for problems like MST and shortest paths.	L3
CO-4	Students will be able to design and implement dynamic programming solutions for problems like TSP and LCS.	L6
CO-5	Students will be able to apply backtracking and graph traversal algorithms like DFS and BFS effectively.	L3
CO-6	Students will be able to explain the significance of NP-Completeness and identify NP-Hard problems.	L2

Unit I: Introduction to Algorithm **(6 Hrs.)**

Introduction: Algorithm definition, Properties of Algorithms, Performance analysis - Space complexity, Time complexity, Algorithm Design Technique, Asymptotic Notation- big (O) notation, omega notation, theta notation.

Unit II: Divide and Conquer **(6 Hrs.)**

Divide and Conquer: Introduction to Divide and Conquer Technique, Binary search, Merge sort, Quick sort, Strassen's matrix multiplication.

Unit III: Greedy Methods **(6 Hrs.)**

Greedy Methods: Introduction, Knapsack Problem, Job sequencing with deadlines, Minimum Spanning Trees, Prim's Algorithms, Kruskal's Algorithm, Dijkstras Shortest Path Algorithm.

Unit IV: Dynamic Programming **(6 Hrs.)**

Dynamic Programming: Introduction, Multistage Graphs, Traveling Salesman, Matrix multiplication, Longest Common Sub-Sequences, Single Source Shortest Paths.

Unit V: Backtracking, Graphs **(6 Hrs.)**

Backtracking: Backtracking Strategy, N-Queen's problem: 4-Queen Problem, 8-Queen Problem, Graph coloring, Hamiltonian Cycle. Graphs: Traversing Graphs, Depth First Search, Breath First Search.

Unit VI: NP-Completeness **(6 Hrs.)**

Introduction to NP-Completeness: The class P and NP, Polynomial time reduction, NP- Completeness Problem, NP-Hard Problems.

Total Lecture 36 Hours

Textbooks:

1. A.A. Puntambekar: "Design and Analysis of Algorithms", Technical Publications.
2. Rajesh K. Shukla: "Design and Analysis of Algorithms", Wiley India Pvt. Ltd..

Reference Books:

1. Dave and Dave: "Design and Analysis of Algorithms" Pearson Education
2. G. Brassard, P.Bratley: "Fundamentals of Algorithmics", PHI
3. Horowitz & Sahani: "Fundamental Algorithms", Galgotia.
4. Cormen, T.H, Lierson & Rivest: "Introduction to Algorithms", Mc Graw-Hill

Program: B.Tech.(IT)				Semester: VI		
Course: Design & Analysis of Algorithm Lab				Code: BTITPC21IT6P		
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

1. To analyze the efficiency of algorithms using asymptotic notations and evaluate their performance.
2. To understand and compare Strassen's method with conventional matrix multiplication based on computational complexity.
3. To develop optimization strategies by comparing dynamic programming and greedy methods through problems like the Knapsack problem.
4. To implement Kruskal's and Prim's algorithms for Minimum Spanning Tree using sorting and union-find techniques.
5. To understand and apply Dijkstra's algorithm for shortest path problems using graph traversal and priority queues.
6. To implement graph traversal and path-finding algorithms for solving complex graph-based problems efficiently.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO1	Analyze and demonstrate the asymptotic performance of algorithms.	L2, L4
CO2	Understand and compare the performance of Strassen's method with the conventional matrix multiplication algorithm.	L2
CO3	Compare the performance of dynamic programming and greedy strategies in solving the Knapsack problem.	L5
CO4	Understand and implement Kruskal's and Prim's Algorithm for finding the Minimum Spanning Tree (MST) of a graph, while developing problem-solving skills related to graph theory, sorting, and the union-find data structure.	L3
CO5	Understand and implement Dijkstra's Algorithm for finding the shortest path in a weighted graph, while gaining proficiency in graph traversal techniques and priority queue data structures.	L3
CO6	Understand and implement graph traversal techniques and path finding algorithms.	L2

General Guidelines: The Provided Sample List of Practical is intended as a guide, and the subject teacher has the flexibility to customized practical list aligned with the DA curriculum.

Pract. No.	List of Practicals
1	Introduction to Design and Analysis of Algorithms.

2	Write a program to study & implement concept of Strassen's Multiplication of 2*2 matrices.
3	Write a program to study & implement concept of Knapsack Algorithm.
4	Write a program to study & implement concept of Kruskal's Algorithm.
5	Write a program to study & implement concept of Prim's Algorithm.
6	Write a program to study & implement concept of Dijkstra's Algorithm.
7	Write a program to study & implement concept of Longest Common Subsequence.
8	Write a program to study & implement concept of Backtracking using Depth First Search.

Program:	B. Tech. (Information Technology)			Semester:				VI
Course:	Cryptography and Network Security			Code:				BTITPC22IT6T
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignment/Quizzes, Attendance, Viva-Voce

Course Objectives:

- Understand Security Concepts.
- Know about various encryption techniques.
- Understand the concept of public key cryptography.
- Study about message authentication and hash functions.
- Impart knowledge on Network security, Internet Security Protocols.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Classify the symmetric encryption techniques	L2
CO-2	Illustrate various public key cryptographic techniques	L2
CO-3	Evaluate the authentication and hash algorithms.	L5
CO-4	Discuss authentication applications	L2
CO-5	Summarize the intrusion detection and its solutions to overcome the attacks.	L2
CO-6	Understand basic concepts of system level security	L1

Unit I: Fundamentals of Security and Encryption **(6 Hrs.)**

Attacks on Computers and Computer Security: Introduction, Need for Security, Security Approaches, Principles of Security, Types of Attacks. Cryptography: Concepts and Techniques Introduction, Plain Text and Cipher Text, Substitution and Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric Key Cryptography, Stenography, Key Range and Key Size, Possible Types of Attacks

Unit II: Symmetric Encryption Algorithms and AES **(6 Hrs.)**

Symmetric Key Algorithms and AES: Introduction, Algorithm Types and Modes, An Overview of Symmetric Key Cryptography, Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), RC4, RC5, Blowfish, Advanced Encryption Standard (AES).

Unit III: Asymmetric Encryption **(6 Hrs.)**

Asymmetric Key Algorithms, Digital Signatures and RSA: Introduction, History and Overview of Asymmetric Key Cryptography, The RSA Algorithm, Symmetric and Asymmetric Cryptography, Digital Signatures, Knapsack and other Algorithms.

Unit IV: Digital Certificates and PKI **(6 Hrs.)**

Digital Certificates and Public Key Infrastructure (PKI): Introduction, Digital Certificates, Private Key Management, The PKIX Model, Public Key Cryptography Standards (PKCS), XML, PKI and Security, Creating Digital Certificate.

Unit V: Internet Security Protocols **(6 Hrs.)**

Internet Security Protocols: Introduction, Concepts, Secure Socket Layer (SSL), Transport Layer security (TLS), Secure Hypertext Transport Protocol (SHTTP), Time Stamping Protocol (TSP), Secure Electronic Transaction (SET), SSL Versus SET, 3-D Secure Protocol, Electronic Money, Email Security, Wireless Application Protocol (WAP)Security, Security in GSM, Security in 3G.

Unit VI: User Authentication Mechanisms **(6 Hrs.)**

User Authentication and Kerberos: Introduction, Authentication Basics, Passwords, Authentication Tokens, Certificate-based-Authentication, Biometric Authentication, Kerberos, Key Distribution Center (KDC), Security Handshake Pitfalls, Single Sign On (SSO) Approaches.

Total Lecture 36 Hours

Textbooks:

1. Atul Kahate, "Cryptography and Network Security", McGraw Hill, Second Edition.

Reference Books:

1. William Stallings, "Cryptography and Network Security, Principles and Practice", PHI Fourth Edition.
2. Behrouz A. Forouzan and Debdeep Mukhopadhyay, "Cryptography and Network Security", McGraw Hill, Second Edition.

MOOCs Links and additional reading, learning, video material

1. Link: NPTEL Cryptography and Network Security
2. Link: Coursera Cybersecurity Specialization

Program:	B.Tech. (IT)			Semester:	VI	
Course:	Cryptography & Network security lab			Code:	BTITPC23IT6P	
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

- To understand basic security threats and protection techniques.
- To implement encryption and decryption using cryptographic methods.
- To work with symmetric key algorithms like DES and AES.
- To apply RSA and digital signatures using public key cryptography.
- To create and analyse digital certificates and PKI.
- To explore authentication methods and simulate Kerberos protocol.

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

CO	Course Outcomes	BT Level (L1 to L6)
1.	Identify and classify computer security threats and attack types.	L2
2.	Implement basic encryption and decryption techniques.	L3
3.	Apply symmetric encryption algorithms like DES and AES.	L3
4.	Use RSA and digital signatures for secure communication.	L4
5.	Generate and verify digital certificates using tools.	L4
6.	Demonstrate authentication protocols like Kerberos and SSO.	L3

Expt. No.	List of Experiments
1	Study types of computer security attacks and protection methods.
2	Implement Caesar Cipher (Substitution) and Rail Fence (Transposition) techniques.
3	Perform encryption and decryption using DES algorithm.
4	Perform encryption and decryption using AES algorithm.
5	Implement RC4 or RC5 stream cipher algorithm.
6	Implement RSA algorithm for encryption and decryption.
7	Generate and verify digital signatures.

8	Create a self-signed digital certificate using Open SSL.
9	Capture and analyze SSL/TLS handshake using Wireshark.
10	Simulate the Kerberos authentication protocol and understand SSO.

Program:	B.Tech.(IT)			Semester:	VI			
Course:	Big Data Analytics			Code:	BTITPE07IT6T			
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE- II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignment/Quizzes, Attendance, Viva-Voce

Course Objectives:

1. To make the students aware about the basic concepts of big data analytics.
2. To introduce the tools required to manage and analyze big data like Hadoop and NoSQL
3. To discuss the basic concepts and operations of map-Reduce
4. To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
5. To introduce several new algorithms for big data mining like classification, clustering and finding frequent patterns
6. To introduce to the students several types of big data like social media, Healthcare and data streams and help them to solve real world problems in for decision support.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Understand the key issues in big data management and its associated applications in intelligent business and scientific computing.	L1
CO-2	Acquire fundamental enabling techniques like Hadoop, and NO SQL in big data analytics.	L2
CO-3	Achieve basic knowledge and operations of Map-Reduce	L2
CO-4	Interpret business models and scientific computing paradigms and apply software tools for big data analytics.	L4
CO-5	Implement algorithms for Clustering, Classifying and finding associations in Big Data	L4
CO-6	Achieve adequate perspectives of big data analytics in various applications like healthcare and social media applications.	L3

Unit I: Introduction to Big Data **(6 Hrs.)**

Definition of Big Data, Types of Data, Big data characteristics & considerations, Challenges of Big Data, Traditional vs. Big Data business approach, Case Study of Big Data Solutions.

Unit II: Introduction to big data frameworks: Hadoop and NoSQL **(6 Hrs.)**

Introduction to Hadoop, Hadoop Components; Hadoop Ecosystem; Overview : Apache Spark, Pig, Hive, HBase, Sqoop, Introduction to NoSQL, NoSQL data architecture patterns: Key-value stores, Graph stores, Column family (Bigtable) stores, Document stores, Mongo DB.

Unit III: MapReduce Paradigm **(6 Hrs.)**

MapReduce: The Map Tasks, Grouping by Key, The Reduce Tasks, Combiners, Details of MapReduce Execution, Coping with Node Failures. Algorithms Using MapReduce: Matrix-Vector Multiplication by MapReduce, Relational-Algebra Operations, Matrix Multiplication with Map Reduce.

Unit IV: Mining Big Data Stream **(6 Hrs.)**

The Stream Data Model: A DataStream-Management System, Examples of Stream Sources, Stream Queries, Sampling Data in a Stream: Sampling Techniques. Filtering Streams: The Bloom Filter.
Counting Distinct Elements in a Stream: The Flajolet-Martin Algorithm, Counting Ones in a Window: The Datar-Gionis- Indyk Motwani Algorithm, Query Answering in the DGIM Algorithm.

Unit V: Big Data Mining Algorithms **(6 Hrs.)**

Frequent Pattern Mining: Park, Chen, and Yu Algorithm, SON Algorithm and MapReduce. Clustering Algorithms: CURE Algorithm. Canopy Clustering, Clustering with MapReduce. Classification Algorithms: Parallel Decision trees, Overview SVM classifiers.

Unit VI: Big Data Analytics Applications **(6 Hrs.)**

Application of Big Data in Enterprises, Application of IoT Based Big Data, Application of Online Social Network-Oriented Big Data, Applications of Healthcare and Medical Big Data, Collective Intelligence, Smart Grid.

Total Lecture 36 Hours

Textbooks:

1. Radha Shankarmani, M Vijayalakshmi," Big Data Analytics", Wiley Publications
2. Anand Rajaraman and Jeff Ullman "Mining of Massive Datasets", Cambridge University Press
3. Christopher D. Manning and Hinrich Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Reference Books:

1. Analytics in a Big Data World: The Essential Guide to Data Science and its Applications, Bart Baesens, WILEY Big Data Series
2. Hadoop: The Definitive Guide by Tom White, O'Reilly Publications.
3. Big Data Analytics with R and Hadoop by Vignesh Prajapati Paperback, Packt Publishing Limited

MOOCs Links and additional reading, learning, video material

1. https://onlinecourses.nptel.ac.in/noc20_cs92/preview
2. https://onlinecourses.swayam2.ac.in/arp19_ap60/preview
3. https://www.edureka.co/blog/big-data-tutorial

Program:	B.Tech.(IT)			Semester:		VI
Course:	Big Data Analytics Lab			Code:		BTITPE08IT6P
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

- To understand and compare different big data analytics tools and their applications.
- To introduce the R programming environment and guide through its installation process.
- To learn the basics of R programming, including variable declaration, expressions, and functions.
- To understand the creation and manipulation of lists in R, including merging lists, and adding matrices and vectors to lists
- To gain proficiency in data manipulation and processing in R, including merging datasets, sorting, plotting, and managing data using matrices and data frames.
- To perform text and Sentiment analysis on datasets (e.g., Iris, Painters, or Covid) using R, understanding the steps involved in processing and assess the sentiment trends within the dataset

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

CO	Course Outcomes	BT Level (L1 to L6)
CO1	Identify and describe the functionalities of various big data analytics tools, and evaluate their suitability for different types of data analysis tasks.	L2
CO2	Implement the Hadoop Map Reduce application for counting frequency of words	L2
CO3	Install R language and understand its basic Operations to perform basic computations.	L2
CO4	Create lists in R, merge multiple lists, and perform operations such as adding matrices and vectors within lists.	L3
CO5	Manipulate and process data in R by Creating Data Frames, merging datasets, sorting data, creating plots, and managing data using matrices and data frames.	L3
CO6	Perform text and Sentiment analysis on datasets using R, interpret the results, and draw meaningful insights.	L4

General Guidelines: The Provided Sample List of Practical is intended as a guide and the subject teacher has the flexibility to customized practical list aligned with the BDA curriculum.

Pract. No.	List of Practicals
1	Study of various Big data Analytics tools.
2	Study of Hadoop MapReduce Application for counting frequency of words/phrase from a simple text file using PySpark

3	Introduction and Installation of R
4	Study of R: Declaring Variable, Expression, Function and Executing R
5	Creating List in R – merging two lists, adding matrices in lists, adding vectors in list
6	Manipulating & Processing Data in R – merging data sets, sorting data, plotting data, managing data using matrices & data frames
7	Text Analysis using R on Iris dataset
8	Text Analysis using R on Painters/Covid dataset
9	Sentiment Analysis in R using built in dataset/Dataset available in Kaggle.

Program:	B.Tech.(IT)			Semester:	VI			
Course:	Program Elective I-Computer Vision			Code:	BTITPE09IT6T			
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE- II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignment/Quizzes, Attendance, Viva-Voce

Course Objectives:

1. To introduce fundamental concepts and techniques used in computer vision.
2. To understand image formation, preprocessing, feature extraction, and segmentation.
3. To implement object recognition and motion tracking algorithms.
4. To apply computer vision in real-life applications like surveillance, biometrics, and robotics.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Understand image acquisition and camera modeling techniques.	L2
CO-2	Apply spatial and frequency domain techniques for image enhancement.	L3
CO-3	Extract and match key features in digital images.	L3
CO-4	Perform object segmentation and shape analysis.	L4
CO-5	Recognize and classify objects using various recognition methods.	L4
CO-6	Analyze motion and apply vision techniques to practical scenarios.	L5

Unit I: Introduction to computer vision **(6 Hrs.)**

Introduction to computer vision, Applications, Human vs machine vision, Image acquisition, Light and radiometry, Camera geometry and calibration, Pinhole camera model, Perspective projection

Unit II: Image Preprocessing and Enhancement **(6 Hrs.)**

Image digitization, Histogram operations, Point processing, Filtering: linear (mean, Gaussian), non-linear (median), Edge detection (Sobel, Prewitt, Canny), Noise models and smoothing.

Unit III: Feature Detection and Matching **(6 Hrs.)**

Corner detection (Harris), Edge linking, Blob detection, Hough transform, Feature descriptors (SIFT overview), Matching techniques, Geometric alignment

Unit IV: Segmentation and Morphological Processing **(6 Hrs.)**

Thresholding (simple, adaptive), Region growing, Region splitting and merging, Watershed algorithm, Morphological operations: erosion, dilation, opening, closing, shape descriptors.

Unit V: Object Re cognition and Classification **(6 Hrs.)**

Object models, Matching by shape and appearance, Template matching, Feature-based recognition, k-NN, SVM (basic concept), Neural networks (introductory), Training and testing models

Unit VI: Motion Analysis and Vision Applications **(6 Hrs.)**

Optical flow, Frame differencing, Background subtraction, Kalman filter basics, Object tracking, Face detection, Gesture recognition, Applications in robotics, medical, surveillance

Total Lecture **36 Hours**

Textbooks:

1. **E. R. Davies**, *Computer Vision: Principles, Algorithms, Applications, Learning* Publisher: Academic Press / Elsevier, Indian edition available ISBN: 9788131221777
2. **Deepak Garg and K. S. Ray**, *Computer Vision*, Khanna Publishing House

Reference Books:

1. **A. Ravishankar Rao** *Computer Vision: A Modern Approach* Pearson Education India (Indian adaptation of Forsyth & Ponce)
2. **Sonka, Hlavac, and Boyle** *Image Processing, Analysis, and Machine Vision* Cengage Learning (Indian edition available)

MOOCs Links and additional reading, learning, video material

1. https://onlinecourses.nptel.ac.in/noc25_cs143/preview
2. https://onlinecourses.nptel.ac.in/noc25_cs93/preview

Program:	B.Tech. (Information Technology)			Semester:	VI	
Course:	PE-I Computer Vision Lab			Code:	BTITPE10IT6P	
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	0	2	1	30	20	50

Course Objectives: To enable students to design, develop, deploy, and secure cloud-based applications using cloud architecture, service models, virtualization, and open-source or public cloud platforms through a structured mini-project approach.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO1	Implement basic image processing operations such as reading, writing, transformation, and enhancement techniques.	L3
CO2	Perform and analyze image enhancement techniques using histogram equalization and filtering methods.	L4
CO3	Apply Fourier Transform for frequency-domain analysis and filtering of images.	L3
CO4	Extract image features using SIFT and HOG and analyze their effectiveness in image understanding.	L4
CO5	Implement and evaluate segmentation and motion analysis techniques such as optical flow.	L5
CO6	Design and develop a mini-project using classification or object recognition techniques on real-world datasets	L6

List of Experiment

1. Implementing various basic image processing operations in python/MATLAB/open-CV: Reading image, writing image, conversion of images, and complement of an image
2. Implement contrast adjustment of an image. Implement Histogram processing and equalization.
3. Implement the various low pass and high pass filtering mechanisms.
4. Use of Fourier transform for filtering the image.
5. Utilization of SIFT and HOG features for image analysis.
6. Performing/Implementing image segmentation
7. Implement optical flow computation algorithm.
8. Demonstrate the use of optical flow in any image processing application
9. Object detection and Recognition on available online image datasets
10. Character or digit or face classification project.

Program:	B. Tech. (Information Technology)			Semester:			VI	
Course:	Intermediate Java Programming			Code:			BTITPE11IT6T	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	30	30	60	100

Methods of Teacher Assessment (TA): Continuous Assessment (40%), Assignment/Quizzes (40%), Attendance (20%)

Course Objectives:

- To strengthen object-oriented thinking through advanced class structuring and basic design patterns.
- To apply generics and utility classes for writing reusable, efficient, and type-safe code.
- To enable data persistence and input validation using serialization, JSON, and regular expressions.
- To develop database connectivity skills using JDBC for building data-driven Java applications.
- To design user-friendly desktop applications using Java Swing and JavaFX.
- To implement basic networking concepts and perform unit testing for reliable application development.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Design well-structured **Java classes** using **OOP concepts** and apply basic **design patterns**	L3
CO-2	Utilize **generics**, **Collections**, and **utility classes** to implement **custom sorting** and operations	L3
CO-3	Perform **serialization**, generate and parse **JSON**, and validate input using **regular expressions**	L4
CO-4	Develop **Java applications** that interact with **relational databases** using **JDBC** and **DAO**	L5
CO-5	Build **desktop applications** using **Java Swing/JavaFX** with form design and **event-driven programming**	L3
CO-6	Implement basic **networking applications** and write **JUnit test cases** for **code validation**	L4

Unit I: Class Structuring & Design Patterns **(6 Hrs.)**

Design principles (DRY, KISS, SOLID), object immutability, static blocks and methods, nested classes (static, non-static, anonymous), enum types, record classes, access modifiers, intro to Singleton and Factory design patterns.

Unit II: Generics, Utility Classes & Object Sorting **(6 Hrs.)**

Generic classes and methods, bounded types, wildcards (?, extends, super), Collections utility class methods (sort, shuffle, binarySearch, reverse), Arrays utility methods, Objects and Optional classes, implementing Comparable interface, using Comparator for custom sorting, sorting lists of custom objects.

Unit III: Serialization, JSON Handling & Regular Expressions **(6 Hrs.)**

Serializable interface, transient keyword, ObjectInputStream, ObjectOutputStream, JSON parsing and generation using Gson/org.json, Pattern and Matcher classes, input validation using regex (email, password, phone), regex for text filtering and replacement

Unit IV: JDBC – Java Database Connectivity **(6 Hrs.)**

JDBC architecture, types of JDBC drivers, establishing database connections with MySQL or PostgreSQL, using Connection, executing queries using Statement and PreparedStatement, handling results with ResultSet, performing basic CRUD (Create, Read, Update, Delete) operations, exception handling in database operations, introduction to DAO (Data Access Object) design pattern.

Unit V: GUI Programming – Java Swing & JavaFX **(6 Hrs.)**

Creating windows using JFrame, adding form elements using JLabel, JTextField, and JButton, using FlowLayout and GridLayout for form alignment, handling button click events using ActionListener, reading input from text fields and displaying output using JOptionPane, building a basic login or registration form using Swing.

JavaFX introduction: Stage, Scene, VBox, HBox, using Label, TextField, and Button controls, setting up basic UI layout using GridPane, handling button clicks using setOnAction method, showing output in JavaFX Label or Alert dialog.

Unit VI: Networking & JUnit Testing **(6 Hrs.)**

Networking fundamentals, InetAddress class, Socket and ServerSocket classes, URL and URLConnection classes, HttpURLConnection class, client-server communication basics, intro to REST, unit testing using JUnit, test cases, assertions, annotations (@Test, @Before, @After), intro to Maven for test execution

Total Lecture 36 Hours

Textbooks:

1. **Cay S. Horstmann.** *"Core Java Volume I – Fundamentals"*, Pearson Education, 11th Edition, 2018.
2. **Herbert Schildt.** *"Java: The Complete Reference"*, McGraw Hill Education, 11th Edition, 2019.

Reference Books:

1. Jim Keogh."J2EE: The complete Reference Paperback" McGraw Hill Education. 1st edition,2017
2. Y. Daniel Liang. *"Introduction to Java Programming and Data Structures"*, Pearson Education, 12th Edition, 2021.

MOOCs Links and additional reading, learning, video material

1. **Coursera**: Java Programming and Software Engineering Fundamentals Specialization
   Link – https://www.coursera.org/specializations/java-programming
2. **Oracle's Java Tutorials**
   Link - https://docs.oracle.com/javase/tutorial/
3. **Book: "Head First Java" by Kathy Sierra and Bert Bates**
   Link - https://www.amazon.com/Head-First-Java-Kathy-Sierra/dp/0596009208

Program:	B. Tech. (Information Technology)			Semester:		VI
Course:	Intermediate Java Programming Lab			Code:		BTITPE12IT6P
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

- To strengthen understanding of **object-oriented programming** and explore advanced **class structuring** and **design patterns** through practical coding tasks.
- To apply **generics, collections**, and **sorting techniques** in developing reusable Java solutions.
- To implement **data persistence, serialization,** and **JSON processing** for handling structured and unstructured data.
- To develop **database-driven applications** using **JDBC** and organize access logic using the **DAO design pattern**.
- To build interactive **GUI-based applications** using **Java Swing/JavaFX** with form validation and event-driven programming.
- To apply **networking concepts** and develop **unit-tested, thread-safe** applications using multithreading and JUnit.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	**Demonstrate** use of **OOP principles**, static blocks, nested classes, and apply design patterns	L3
CO-2	**Implement** generic methods and utilize **Collections API** for data sorting and processing	L3
CO-3	**Apply** object **serialization** and **JSON conversion** for persistent data handling	L4
CO-4	**Build** JDBC applications and **evaluate** DAO design pattern for modular database logic	L5
CO-5	**Design and develop** desktop apps with forms, layout, and validation using **Swing/JavaFX**	L6
CO-6	**Develop** socket-based applications and **test** program logic using **JUnit**	L4

General Guidelines: The provided sample list of programs is intended as a guide, and the subject teacher has the flexibility to curate a customized set of practical's aligned with the curriculum.

Sr. No.	List of Practical
1	Create a program demonstrating static, non-static, and anonymous inner classes, along with enums.
2	Implement the Singleton design pattern and show that only one object can be created.
3	Create a factory class to generate different types of employee objects (e.g., FullTime, Intern).
4	Write a generic method that reverses elements of any array type.
5	Sort a list of products based on price using the Comparable interface.
6	Sort employee records by name and then by department using Comparator.
7	Demonstrate the use of Collections methods: sort(), shuffle(), binarySearch().
8	Use Optional to safely retrieve values from objects that may be null.
9	Serialize a Student object and save it to a file, then read it back.
10	Use Gson or org.json to convert an object to JSON and parse it back.
11	Validate email ID, phone number, and password strength using regular expressions.
12	Read a text log file and extract date, time, and error messages using regex.
13	Write a program to establish a connection with MySQL or PostgreSQL.
14	Create a console app that performs Insert, Update, Delete, and View operations.
15	Demonstrate commit and rollback using manual transaction control.
16	Design a DAO class to separate business logic from data access logic.
17	Create a login form with username and password fields, and show success/failure.
18	Design a registration form using JavaFX with text fields and a submit button.
19	Create a form in Swing that accepts user input and displays data in a JTable.
20	Build a simple client and server program for message exchange.
21	Write JUnit test cases for a Calculator or utility class (add, subtract, divide).

(Develop any 1, sample reference given below)

Mini Project 1 - *Create a system with GUI to add, view, update, and delete student records using JDBC.*
Concepts: DAO, JDBC, Swing or JavaFX, MVC pattern

Mini Project 2 - *Build a client-server messaging app using sockets and multithreading.*
Concepts: Networking, Threads, Swing

Mini Project 3 - *Track daily expenses with file saving, category filtering, and data summary.*
Concepts: File I/O, GUI, JSON

Program:	B.Tech.(IT)			Semester:	VI			
Course:	Foundation of Machine Learning			Code:	BTITPE13IT6T			
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE- II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA):

Course Objectives:

1. Understand the fundamental concepts of Machine Learning and implement basic ML models using Python.
2. Apply supervised learning algorithms for classification and regression problems and analyze issues such as overfitting, underfitting, and generalization.
3. Demonstrate preprocessing techniques including scaling, transformation, and dimensionality reduction to improve model performance.
4. Implement and evaluate clustering algorithms such as k-Means, Agglomerative Clustering, and DBSCAN for unsupervised learning tasks.
5. Apply feature engineering techniques including categorical encoding, discretization, polynomial features, and feature selection for effective model building.
6. Evaluate and improve machine learning models using cross-validation strategies and appropriate performance metrics for binary classification.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Explain the fundamental concepts of Machine Learning and implement basic ML models using Python libraries.	L1
CO-2	Apply supervised learning algorithms to classification and regression problems and analyze issues such as overfitting and underfitting.	L2
CO-3	Implement preprocessing techniques and dimensionality reduction methods to prepare data for machine learning models.	L3
CO-4	Implement clustering algorithms and analyze their performance on real-world datasets.	L4
CO-5	Analyze and engineer features to improve model representation and predictive performance.	L5
CO-6	Evaluate and compare machine learning models using cross-validation techniques and appropriate performance metrics.	L6

Unit I: Introduction to Machine Learning **(6 Hrs.)**

Importance of Machine Learning, Need of Python, Essential Libraries and Tools, Classifying Iris Species: - Meet the Data, Measuring Success, Training and Testing Data, Building Your First Model: k-Nearest Neighbors, Making Predictions, Evaluating the Model.

Unit II: Supervised Learning **(6 Hrs.)**

Classification and Regression, Generalization, Overfitting, and Underfitting:- Relation of Model Complexity to Dataset Size, Supervised Machine Learning Algorithms:- sample dataset, k-Nearest Neighbors, k-Neighbors classification, Analyzing K-Neighbors Classifier, Linear Models: - Linear models for regression, Naive Bayes Classifiers, Strengths, weaknesses, and parameters.

Unit III: Unsupervised Learning and Preprocessing **(6 Hrs.)**

Types of Unsupervised Learning, Challenges in Unsupervised Learning, Preprocessing and Scaling: - Different Kinds of Preprocessing, Applying Data Transformations, Scaling Training and Test Data the Same Way, The Effect of Preprocessing on Supervised Learning, Dimensionality Reduction, Feature Extraction, and Manifold Learning: - Principal Component Analysis (PCA).

Unit IV: Clustering **(6 Hrs.)**

k-Means Clustering:- Failure cases of k-means, Vector quantization, or seeing k-means as decomposition, Agglomerative Clustering:-Hierarchical clustering and dendrograms, DBSCAN.

Unit V: Representing Data and Engineering Features **(6 Hrs.)**

Categorical Variables, Binning, Discretization, Linear Models, and Trees, Interactions and Polynomials, Univariate Nonlinear Transformations, Automatic Feature Selection.

Unit VI: Model Evaluation and Improvement **(6 Hrs.)**

Cross-Validation:- Cross-Validation in scikit-learn, Benefits of Cross-Validation, Stratified k-Fold Cross-Validation and Other Strategies, Evaluation Metrics and Scoring:-Metrics for Binary Classification.

Total Lecture 36 Hours

Textbooks:

1. Introduction to Machine Learning with Python- Andreas C. Müller & Sarah Guido.

Reference Books:

1. The Hundred-Page Machine Learning Book–Andriy Burkov
2. Hands-on Machine Learning with Scikit-Learn, Keras, and Tensor Flow: Concepts, Tools, and Techniques to Build Intelligent Systems –Aurelien Geron
3. Introduction to Machine Learning with Python: A Guide for Data Scientists– Andreas C. Müller & Sarah Guido

4. Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning. –Chris Albon

MOOCs Links and additional reading, learning, video material

1. NPTEL :- Introduction to Machine Learning, IIT Madras, https://nptel.ac.in/courses/106106139
2. Coursera :- Machine Learning Specialization
3. Udemy :- Machine Learning A-Z,

Program: B.Tech.(IT)				Semester: VI		
Course: Foundation of Machine Learning Lab				Code: BTITPE14IT6P		
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

1. Introduce ensemble learning methods to improve classification accuracy.
2. Provide hands-on experience in sentiment analysis using real-world text datasets.
3. Teach deployment of machine learning models using web technologies.
4. Enable learners to apply regression and clustering techniques to structured and unstructured data.
5. Guide students in designing and training neural networks (MLP, CNN, RNN).
6. Develop ability to visualize and debug deep learning models using TensorBoard.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO1	Implement bagging, boosting, and majority voting using scikit-learn.	L3
CO2	Build sentiment classification models using logistic regression and bag-of-words.	L3
CO3	Develop and deploy ML models via Flask and SQLite.	L4
CO4	Apply and evaluate linear and regularized regression models.	L5
CO5	Construct CNN models and interpret learning via TensorBoard.	L4
CO6	Develop RNN models for sequential data including language and sentiment tasks.	L6

General Guidelines: The Provided Sample List of Practical is intended as a guide and the subject teacher has the flexibility to customized practical list aligned with the curriculum.

Pract. No.	List of Practicals
1	Installation and setup of Python, Anaconda/Jupyter Notebook.
2	Introduction to NumPy, Pandas, Matplotlib, and Scikit-learn.
3	Implement k-Nearest Neighbors algorithm.
4	Implement Linear Regression on sample dataset.
5	Implement Gaussian Naïve Bayes classifier
6	Apply StandardScaler and MinMaxScaler
7	Apply StandardScaler and MinMaxScaler
8	Implement K-Means clustering

Program:	B. Tech. (Information Technology)	Semester:	VI
Course:	Program Elective Course II (Blockchain Technology)	Code:	BTITPE15IT6T

Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA): Assignment/Quizzes, Attendance, Viva-Voce

- Understand the architecture, characteristics, and mining mechanisms of decentralized blockchain systems.
- Apply cryptographic fundamentals and differentiate between types of blockchains and their protocols.
- Analyze various distributed consensus algorithms and their role in securing blockchain networks.
- Develop and execute smart contracts using Ethereum and comprehend its architecture and scripting.
- Explore and compare different blockchain platforms and tools used in blockchain development and deployment.
- Evaluate real-world blockchain applications and emerging trends integrating blockchain with advanced technologies.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Understand the architecture, characteristics, and mining mechanisms of decentralized blockchain systems.	L2
CO-2	Apply cryptographic techniques and distinguish between types of blockchains and their associated protocols.	L3
CO-3	Analyze consensus algorithms and their effectiveness in ensuring agreement across distributed systems.	L4
CO-4	Develop and deploy smart contracts on Ethereum while understanding its architecture and scripting mechanisms.	L3
CO-5	Compare various blockchain platforms (e.g., Hyperledger, Corda) and utilize tools for development and deployment.	L4
CO-6	Evaluate real-world blockchain applications and innovations like NFTs, DeFi, AI, IoT, and Cloud integration.	L5

Unit I: Introduction to Blockchain **(6 Hrs.)**

Introduction to decentralized system, history, conceptualization, architectural principles behind Blockchain, characteristics of Blockchain, mining strategy.

Unit II: Basic Crypto Primitives **(6 Hrs.)**

Description: Hashing, public key cryptosystems. Types of Blockchains: private vs public Blockchain, Blockchain protocol and use cases, hash puzzles.

Unit III: Distributed Consensus **(6 Hrs.)**

Description: Consensus approach, consensus elements. Consensus algorithms: Proof of Work, Byzantine General Problem, Proof of Stake, Proof of Elapsed Time, Proof of Activity, Proof of Burn.

Unit IV: Smart Contracts and Ethereum **(6 Hrs.)**

History, purpose, and types of smart contracts. Introduction to Ethereum, Bitcoin vs Ethereum stack. P2P network in Ethereum, consensus in Ethereum, scripts in Ethereum. Developing and executing smart contracts in Ethereum. State and data structure in Ethereum.

Unit V: Blockchain Platforms and Frameworks **(6 Hrs.)**

Overview of prominent blockchain platforms – Hyperledger Fabric, Hyperledger Sawtooth, and R3 Corda. Comparative analysis of public and permissioned blockchain frameworks. Study of tools for blockchain development and deployment, including Truffle, Ganache, MetaMask, and Remix IDE.

Unit VI: Blockchain Applications and Emerging Trends **(6 Hrs.)**

Real-world applications of blockchain in supply chain, healthcare, finance, land records, and voting systems. Introduction to NFTs, DeFi (Decentralized Finance), DAOs (Decentralized Autonomous Organizations). Exploring integration with emerging technologies like IoT, AI, and Cloud. Future scope and challenges in blockchain adoption.

Total Lecture 36 Hours

Textbooks:

1. Artemis Caro, "Blockchain: The Beginners Guide to Understanding the Technology Behind Bitcoin Cryptocurrency", Kevin Wolhuter, 2021, ISBN: 1922590061, 9781922590060
2. Andreas Antonopoulos, "Mastering Bitcoin: Unlocking Digital Cryptocurrencies", O'Reilly Media, Inc.20 ISBN: 9781491954386

Reference Books:

1. Mark Watney, Blockchain for Beginners: The Complete Step by Step Guide to Understanding Blockch Technology", CreateSpace Independent Publishing Platform, 2017, ISBN: 1548766887, 9781548766887
2. Alwyn Bishop, "Blockchain Technology Explained", CreateSpace Independent Publishing Platform, 2018, ISI 9781986273800

MOOCs Links and additional reading, learning, video material

1. NPTEL Course "Introduction to Block Chain Technology & Applications" https://nptel.ac.in/courses/106/104/106104220/
2. NPTEL Course on "Blockchain Architecture & Use Cases" https://nptel.ac.in/courses/106/105/106105184/

Program: B.Tech. (IT)				Semester: V		
Course: Program Elective Course II Lab (Blockchain Technology)				Code: BTITPE16IT6P		
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

- Simulate decentralized peer-to-peer networks and basic blockchain structures using open-source tools.
- Implement cryptographic and hashing concepts through smart contracts using Solidity and Python.
- Differentiate and analyze public vs private blockchains with deployment on testnets like Ganache.
- Demonstrate consensus mechanisms including Proof of Work, Proof of Stake, and the Byzantine Generals Problem.
- Design, deploy, and test smart contracts using Ethereum Remix IDE, MetaMask, Ganache, and Truffle.
- Develop and present real-world blockchain applications and decentralized apps (DApps) with proper documentation.

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	Simulate a decentralized peer-to-peer network and explain blockchain architecture and mining.	L2
CO-2	Implement cryptographic and hashing techniques using Python and Solidity smart contracts.	L3
CO-3	Differentiate between public and private blockchains using testnets and tools like Ganache.	L4
CO-4	Develop and compare blockchain consensus mechanisms like PoW and PoS through simulation.	L5
CO-5	Design, deploy, and test smart contracts using Ethereum Remix, MetaMask, Ganache, and Truffle.	L6
CO-6	Build and demonstrate a real-world DApp and prepare IEEE-format documentation.	L6

General Guidelines: The provided sample list of programs is intended as a guide, and the subject teacher can create a customized set of experiments aligned with the **Blockchain Technology- Lab** programming curriculum.

Practical. No.	List of Practical's
1	Simulate a decentralized peer-to-peer network and demonstrate basic block structure.

2	Write a report on Blockchain architecture and mining using existing open-source tools.
3	Implement a Solidity program demonstrating basic hashing and cryptography concepts.
4	Differentiate between public and private blockchains using testnet or Ganache.
5	Simulate a hash puzzle (Proof of Work) using Python or Solidity.
6	Develop a program to simulate the consensus mechanism (e.g., Proof of Stake vs PoW).
7	Write a script that simulates the Byzantine Generals Problem.
8	Write a smart contract in Solidity for a basic banking application (deposit, withdraw, getBalance).
9	Deploy and test a smart contract on the Ethereum Remix IDE and connect with MetaMask.
10	Implement an event-driven smart contract in Ethereum (e.g., voter registration and result logging).
11	Compare Ethereum and Hyperledger Fabric by creating a basic transaction example in both.
12	Setup Ganache & Truffle to build and deploy a smart contract with testing.
13	Demonstrate any one use case of Blockchain in land records or healthcare with a working demo.
14	Develop a mini DApp integrating Ethereum smart contracts with a frontend (HTML + JS + Web3.js).
15	Mini Project: In groups, build a Blockchain-based solution and deploy it on a testnet; document in IEEE format.

Program:	B.Tech.(IT)			Semester:	VI			
Course:	Natural Language Processing			Code:	BTITPE17IT6T			
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE- II	ESE	Total
3	-	3	3	10	15	15	60	100

Methods of Teacher Assessment (TA):

Course Objectives:

1. To provide a foundational understanding of Natural Language Processing (NLP).
2. To understand and apply foundational techniques in natural language processing, including tokenization, stemming, lemmatization, morphological analysis.
3. To understand and apply various techniques for Part-Of-Speech (POS) tagging.
4. To explore and analyze the principles of lexical semantics, meaning representation, and the relationships among lexemes.
5. To understand and apply techniques for discourse and pragmatic processing.
6. To analyze and evaluate the implementation of NLP applications.

Course Outcomes:

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

CO	Course Outcomes	BT Level (L1 to L6)
CO-1	To explain the fundamental concepts of NLP, analyze the stages and challenges involved in natural language processing	L2
CO-2	Perform tokenization, stemming, and lemmatization, analyze morphological structures, and compute word sequence probabilities using N-gram models.	L3
CO-3	Understand and compare different POS tagging techniques, such as rule-based and Hidden Markov Models (HMM)	L4
CO-4	Describe and differentiate semantic relations such as homonymy, polysemy, and synonymy, utilize lexical resources like WordNet and BabelNet, and analyze semantic ambiguities in natural language.	L4
CO-5	Explain the principles of discourse coherence, perform reference resolution, and implement anaphora resolution techniques using Hobbs and Centering algorithms.	L4
CO-6	Analyze and evaluate the performance and challenges of various NLP applications	L5

Unit I: Introduction to NLP **(6 Hrs.)**

History of NLP, Generic NLP system, levels of NLP, Knowledge in language processing, Ambiguity in Natural language, stages in NLP, challenges of NLP, Applications of NLP

Unit II: Word Analysis **(6 Hrs.)**

Tokenization, Stemming, Lemmatization; Survey of English Morphology, Inflectional Morphology, Derivational Morphology; Regular expression with types
N-gram, Unigram/Bi Gram Language Models, Computing the Probability of Word Sequence.

Unit III: Syntax Analysis **(6 Hrs.)**

Part-Of-Speech tagging (POS); Tag set for English (Upenn Treebank); Difficulties /Challenges in POS tagging; Rule-based, Stochastic and Transformation-based tagging; Generative Model: Hidden Markov Model (HMM Viterbi) for POS tagging.

Unit IV: Semantic Analysis **(6 Hrs.)**

Introduction, meaning representation; Lexical Semantics; Corpus study; Study of Various language dictionaries like WorldNet, Babelnet; Relations among lexemes & their senses -Homonymy, Polysemy, Synonymy, Hyponymy; Semantic Ambiguity.

Unit V: Pragmatic & Discourse Processing **(6 Hrs.)**

Discourse: Reference Resolution, Reference Phenomena, Syntactic & Semantic constraint on coherence; Anaphora Resolution using Hobbs and Cantering Algorithm

Unit VI: Applications of NLP **(6 Hrs.)**

Case studies on (preferable in regional language): Machine translation; Text Summarization; Sentiment analysis; Information retrieval; Question Answering system

Total Lecture 36 Hours

Textbooks:

1. Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Second Edition, Prentice Hall, 2008.
2. Siddiqui and Tiwary U.S., Natural Language Processing and Information Retrieval, Oxford University Press, 2008.
3. Christopher D. Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.

Reference Books:

1. Daniel M Bikel and ImedZitouni — Multilingual natural language processing applications:
2. Nitin Indurkhya and Fred J. Damerau, —Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC Press, 2010.

MOOCs Links and additional reading, learning, video material

1. https://nptel.ac.in/courses/106105158
2. http://www.cse.iitb.ac.in/~cs626-449
3. https://onlinecourses.nptel.ac.in/noc19_cs56/preview

Program: B.Tech. (IT)				Semester: VI		
Course: Natural Language Processing - Lab				Code: BTITPE18IT6P		
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	-	2	1	30	20	50

Course Objectives:

Discuss current capabilities and future trajectories of NLP, implementing core techniques—from morphological analysis to parsing and word-sense disambiguation—while situating them within wider computer-science principles.

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

CO	Course Outcomes	BT Level (L1 to L6)
1.	Explain how tokenization, stop-word removal, stemming, and lemmatization prepare raw text for downstream NLP tasks.	L2
2.	Apply POS tagging, noun-phrase chunking, and CYK/chart parsing in NLTK to derive sentence structure.	L3
3.	Analyze corpus patterns by computing unigrams, bigrams, trigrams, TF-IDF weights, and sentence probabilities.	L4
4.	Evaluate Named Entity Recognition performance with precision-recall-F1 metrics across diverse domains.	L5
5.	Design and implement sentiment-analysis, spam-filtering, and fake-news detection classifiers using appropriate NLP features.	L6
6.	Create an end-to-end, reproducible NLP pipeline integrating preprocessing, parsing, modeling, and evaluation stages.	L6

General Guidelines:

Expt. No.	List of Experiments
1	a. Perform tokenization by word and sentence using nltk. b. Eliminate stopwords using nltk. C. Implement stemming using nltk
2	a. Develop Parts of Speech tagging using nltk. b. Write a program to perform lemmatization using nltk.
3	a. Write a python program for chunking using nltk. b. Apply Named Entity Recognition using NLTK
4	a. Construct Term Frequency (TF) and Inverse Document Frequency (IDF) and compute the TF-IDF values for a given corpus

	b. Analyze and implement CYK Parsing (Cocke-Younger-Kasami) or Chart Parsing for a given grammar and input string.
5	a. Generate unigrams, bigrams and trigrams present in the given corpus. b. Implement the probability of the given statement "This is my cat" by taking the an example corpus into consideration.
6	Use the Stanford named Entity recognizer to extract entities from the documents. Use it programmatically and output for each document which named entities it contains and of which type.
7	Perform sentiment analysis using NLP
8	Develop Spam Filter using NLP
9	Detect Fake News using NLP

Program:	B. Tech. (Information Technology)			Semester:		VI
Course:	Mobile App Development Lab			Code:		BTITVC05IT6P
Teaching Scheme				Evaluation Scheme		
Practical	Tutorial	Hours	Credit	INT	EXT	Total
2	1	3	2	50	-	50

Course Objectives:

- Understand the basic concepts of Dart programming and Flutter framework for mobile application development.
- Apply fundamental widget-based UI building techniques using Flutter to construct simple mobile interfaces.
- Develop a functional cross-platform mobile app using state management and API integration.

General Guidelines: The provided sample list of programs is intended as a guide, and the subject teacher has the flexibility to curate a customized set of practical's aligned with the curriculum.

Sr. No.	List of Practical
1.	Set Up Flutter SDK and Build a "Hello World" App with Hot Reload
2.	Implement Dart Logic for User Data Using Lists, Maps, and Functions
3.	Create a Class-Based Student Model with Inheritance and Method Overriding
4.	Design a Personal Profile Screen Using Column, Row, Image, and Text Widgets
5.	Build a Validated Login Form Using TextField, Form, and ElevatedButton
6.	Construct a Scrollable Product List UI Using ListView and Stack
7.	Implement Multi-Screen Navigation with Data Transfer Between Pages
8.	Create a Stateful Counter App with Dynamic Color and Font Changes
9.	Save User Preferences Locally Using SharedPreferences Plugin
10.	Add Custom Tap, Double Tap, and Long Press Gestures with GestureDetector
11.	Fetch and Parse Weather Data from an Open API Using the http Package
12.	Display Fetched JSON Data in a Custom Card-Based ListView
13.	Integrate url_launcher to Open Web Links and Send Emails from App
14.	Apply App-Wide Theming with Custom Colors, Fonts, and Button Styles
15.	Develop a Multi-Screen To-Do App with Add, Edit, and Delete Functionality

Build any 1 mini project, sample titles are given below: -

1. Create a multi-screen app where users can add, edit, complete, and delete daily tasks. Use local storage (shared_preferences) to save data.

2. Fetch and display real-time weather data using the OpenWeatherMap API. Show temperature, humidity, and city search functionality.
3. Allow users to create and manage student profiles with fields like name, photo, course, and contact. Include list view and edit/delete actions.
4. Build an app where users can write, save, and delete personal notes. Display notes in a scrollable list and allow searching or filtering.

Program:	B.Tech. (Information Technology)			Semester:			VI	
Course:	Modern Indian Language-Marathi			Code:			BTALAE02SH6T	
Teaching Scheme				Evaluation Scheme				
Lecture	Tutorial	Hours	Credit	TA	MSE-I	MSE-II	ESE	Total
1	0	1	1	50	-	-	-	50

Methods of Teacher Assessment (TA): Class test, Assignment, Case studies, Group Discussions.

Course Objectives:

- मराठी साहित्याची अभिरुची वाढवून व्यक्तिमत्व विकास करणे.
- व्यावहारिक मराठी भाषे द्वारा अभिव्यक्ती कौशल्य विकसित करणे.

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

CO	Course Outcomes	BT Levels
CO-1	समृद्ध मराठी साहित्या चे वाचन करून सर्वांगीण व्यक्तिमत्त्व विकास साधतील.	3
CO-2	व्यावहारिक मराठी द्वारा त्याचे अभिव्यक्ती कौशल्य विकसित होईल.	3

Unit I: मराठी साहित्य परंपरा ओळख, प्रकार वैशिष्ट्ये, मूल्यविचार, समृद्ध जीवन अनुभवाची ओळख. 6 Hrs

Unit II: व्यावहारिक मराठी कौशल्य विकास- मुलाखत, अहवाल लेखन, रेजूमी (resumo) इत्यादी. 6 Hrs

Total :12 Hrs

संदर्भ:

१. मराठी भाषेची समृद्धता व सृजनशीलता- डॉ. अतुल जोशी

२.व्यक्तिमत्व विकास आणि भाषा लेखक : गणेश मोकाशी प्रकाशन : स्नेहवर्धन प्रकाशन, पुणे

३.व्यक्तिमत्व विकास आणि भाषा लेखक : तुषार चांदवडकर प्रकाशन : अथर्व प्रकाशन, जळगाव

Approved in Academic Council Meeting 30/3/2026