

Marathwada Mitra Mandal's College of Engineering
Karvenagar, Pune- 52

An Autonomous Institute Affiliated to SPPU

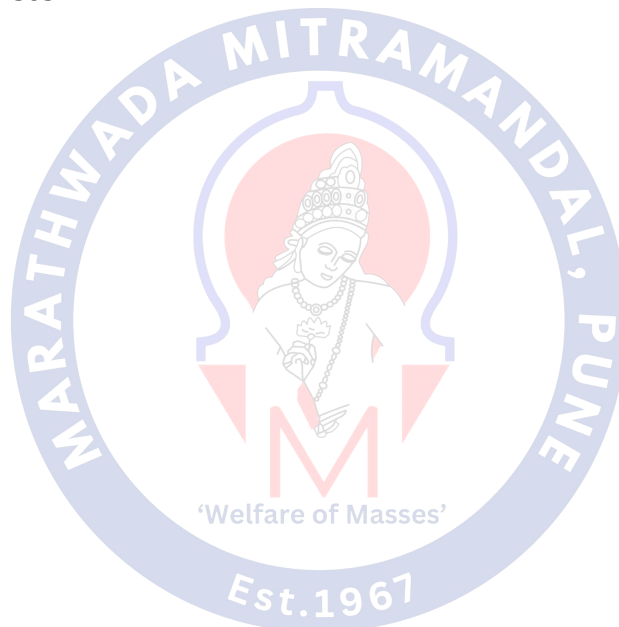


Curriculum for
Third Year B. Tech Information Technology
A.Y. 2026-27

Permanently Affiliated to SPPU | Accredited with A++ grade by NAAC
Recipient of Best College award by SPPU | Accredited by NBA
Recognized under 2(f) and 12(B) of UGC Act 1956
www.mmcoe.edu.in

CONTENTS

Institute Vision and Mission	I
Department Vision and Mission	I
Knowledge and Attitude Profile (WK)	II
Program Outcomes (POs)	III
Program Educational Objectives (PEOs)	IV
Program Specific Outcomes (PSOs)	IV
Abbreviations	V
Curriculum Structure Sem-V	VII
Semester-V Courses	1



Sr. No.	Description	Page
Semester- V		
IT24PCC301	Data Science & Machine Learning	1
IT24PCC302	DevOps Engineering	4
IT24PCC303	Theory of Computation	7
IT24PEC304	A - Project Management	10
	B - Internet of Things	13
	C - Virtualization & Cloud Computing	15
	D - Game Design and Development	18
IT24PCC306	Data Science & Machine Learning Lab	21
IT24PCC307	DevOps Engineering Lab	24
IT24PCC308	Engineering Design and Innovation-I	26
IT24AEC311	Analytical Reasoning and Universal Human Values-I	29



Institute Vision

To be a globally renowned institution through excellence in engineering education for sustainable and holistic development.

Institute Mission

M1: Empower students with cutting-edge technologies and global competencies.

M2: Foster culture of research and entrepreneurial mindset.

M3: Imbibe social and professional values.

M4: Provide an inclusive environment for lifelong learning.

Department Vision

To emerge as a globally competent center in education, research and innovation in Information Technology, fostering sustainable development with a holistic vision.

Department Mission

M1 : To impart IT education that develops globally competent professionals.

M2 : To promote innovation through research and entrepreneurship.

M3 : To encourage lifelong learning and a holistic commitment towards societal development.

Knowledge and Attitude Profile (WK)

- WK1: A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
- WK2: Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
- WK3: A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
- WK4: Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
- WK5: Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
- WK6: Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
- WK7: Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
- WK8: Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
- WK9: Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Program Outcomes (PO)

- **PO1: Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop the solution of complex engineering problems.
- **PO2: Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
- **PO3: Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
- **PO4: Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
- **PO5: Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
- **PO6: The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
- **PO7: Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
- **PO8: Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
- **PO9: Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
- **PO10: Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
- **PO11: Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Program Educational Objectives (PEO)

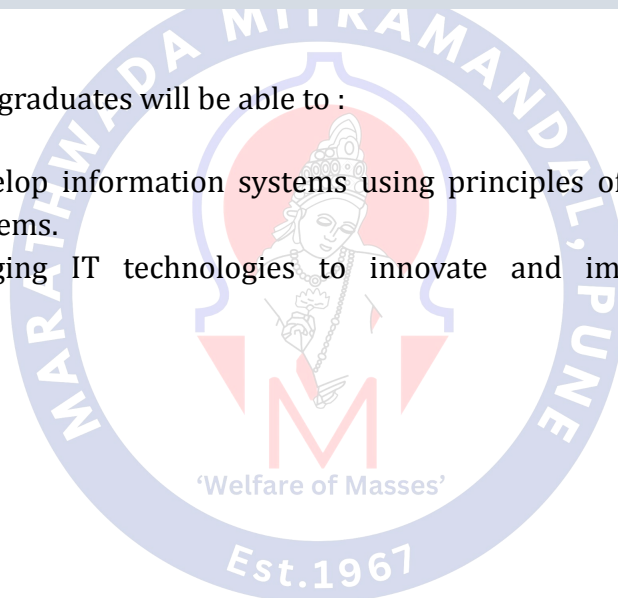
The students of Information Technology Department after passing out will :

- Exhibit the capabilities to excel as IT professionals with strong knowledge and innovative thinking.
- Demonstrate leadership in the IT Industry, pursue lifelong learning and contribute through research & collaboration.
- Apply innovations and ethical practices for sustainable technological advancement and societal enrichment.

Program Specific Outcomes (PSOs)

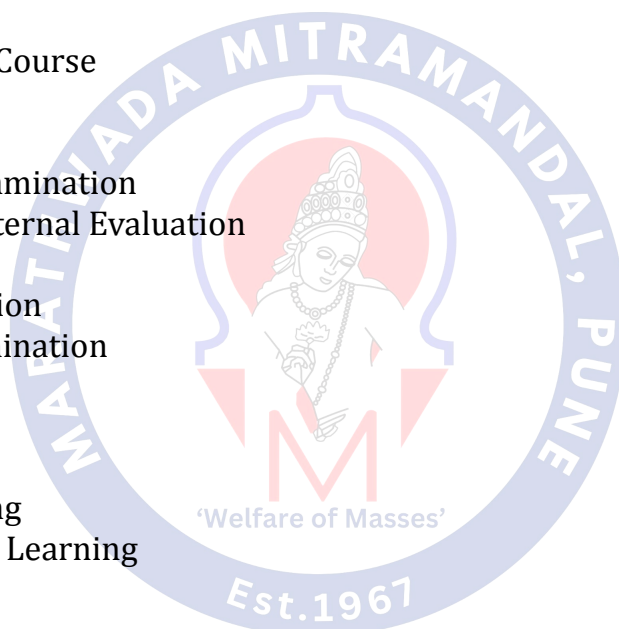
Information Technology graduates will be able to :

- Design and develop information systems using principles of computing to solve real-world problems.
- Integrate emerging IT technologies to innovate and improve technological ecosystems.



Abbreviations

BSC:	Basic Science Course
UG:	Undergraduate Programme
ESC:	Engineering Science Course
PCC:	Program Core Courses
PEC:	Program Elective Courses
MDM:	Multidisciplinary Minor Courses
OE:	Open Elective
VSC:	Vocational Skill Course
SEC:	Skill Enhancement Course
AEC:	Ability Enhancement Course
IKS:	Indian Knowledge System
VEC:	Value Education Course
FP:	Field Project
CC:	Co-Curricular Course
IT:	Internal Tool
ET:	External Tool
ETE:	End -Term Examination
CIE:	Continuous Internal Evaluation
TW:	Term work
OR:	Oral Examination
PR:	Practical Examination
L:	Lecture
P:	Practical
T:	Tutorial
OL:	Online Teaching
ODL:	Open Distance Learning



Curriculum Structure Sem-V

Third Year B.Tech Semester-V																	
Course Code	Course Name	Course Type	Teaching Scheme (Hrs/week)				Examination Scheme						Credits				
			L	P	T	ODL	CIE	ETE	TW	PR	OR	Total	L	P	T	ODL	Total
IT24PCC301	Data Science & Machine Learning	PCC	3	-	-	-	40	60	-	-	-	100	3	-	-	-	3
IT24PCC302	DevOps Engineering	PCC	3	-	-	-	40	60	-	-	-	100	3	-	-	-	3
IT24PCC303	Theory of Computation	PCC	3	-	-	-	40	60	-	-	-	100	3	-	-	-	3
IT24PEC304	A - Project Management B - Internet of Things C-Virtualization & Cloud Computing D-Game Design and Development	PEC	3	-	-	-	40	60	-	-	-	100	3	-	-	-	3
IT24MDM305	*(Multidisciplinary Minor Theory)	MDM	3	-	-	-	40	60	-	-	-	100	3	-	-	-	3
IT24PCC306	Data Science & Machine Learning Lab	PCC	-	2	-	-	-	-	-	50	-	50	-	1	-	-	1
IT24PCC307	DevOps Engineering Lab	PCC	-	2	-	-	-	-	-	50	-	50	-	1	-	-	1
IT24PCC308	Engineering Design and Innovation-I	PCC	-	4	-	-	-	-	50	-	-	50	-	2	-	-	2
IT24MDM309	*(Multidisciplinary Minor Lab)	MDM	-	2	-	-	-	-	25	-	-	25	-	1	-	-	1
IT24OEL310	*(Open Elective Theory)	OEL	2	-	-	-	-	-	50	-	-	50	2	-	-	-	2
IT24AEC311	Analytical Reasoning and Universal Human Values-I	AEC	-	-	1	-	-	-	25	-	-	25	-	-	1	-	1
Total			17	10	1	-	200	300	150	100	-	755	17	5	1	-	23

*** Refer Separate Booklet for MDM and OEL**

Abbreviations:

L- lecture

P- Practical

T- Tutorial

CIE-Continuous Internal Evaluation

ETE- End Term Examination

TW- Term work

PR- Practical

OR- Oral

L : 1 Hr.= 1 credit

P: 2 Hr. = 1 Credit

T: 1 Hr. = 1 Credit



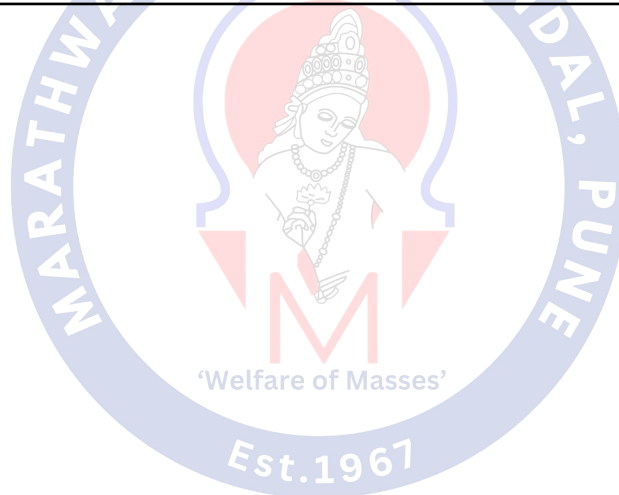
Third Year B. Tech Information Technology														
Semester-V														
Course Code: IT24PCC301					Course Name: Data Science & Machine Learning									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisites: Python programming concepts, statistics and probability														
Course Objectives: - To understand the fundamentals of Data Science, its lifecycle, applications, and ethical considerations. - To perform data collection, cleaning, transformation, and exploratory data analysis using statistical and visualization techniques. - To apply probability, statistical foundations, and feature engineering techniques for data-driven modeling. - To build and evaluate regression, classification, clustering, and probabilistic models using appropriate performance metrics. - To understand the end-to-end machine learning lifecycle and the role of MLOps in deploying, monitoring, and maintaining models.														
Course Outcomes: After completing the course, the students will be able to: C01: Explain fundamentals of machine learning and data preparation. C02: Apply regression models and evaluation metrics. C03: Implement and evaluate classification models. C04: Apply tree-based and probabilistic models. C05: Apply clustering and distance-based learning and explain basic MLOps concepts.														
Unit	Contents													Hrs.
1	Data Science and Exploratory Data Analysis Data Science: Definition, scope, and applications, Data Science lifecycle, Types of data and data sources, Data collection methods, Data ethics, privacy, and security. Data cleaning and handling missing values, Data transformation and normalization, Exploratory Data Analysis, EDA tools & libraries, Descriptive statistics, outlier detection techniques. Case study on real-world datasets													7
2	Machine Learning and Statistical Foundations Types of learning – supervised, unsupervised, semi-supervised, reinforcement; predictive vs descriptive.													9

	Probability distributions: Normal & Binomial; Covariance vs correlation, Mean, median, mode, variance, standard deviation, skewness, and visualization using histograms, boxplots, scatterplots. Feature Engineering: Feature types, construction, transformation, selection using correlation. Sampling and Validation: Sampling methods, training vs testing split, cross-validation techniques – hold-out, k-fold, LOOCV	
3	Regression and Classification Regression Basics, Univariate and multivariate regression; Error metrics – MSE, MAE, R^2 . Polynomial Regression, Underfitting vs overfitting, bias-variance tradeoff. Gradient descent for minimizing cost functions, Ridge & Lasso regression, Linear classification, Evaluation Metrics: Confusion matrix, accuracy, precision, recall, F1-score, ROC-AUC; Multiclass Classification: One-vs-One, One-vs-Rest, weighted metrics; Logistic Regression - Sigmoid function, Support Vector Machines	8
4	Tree and Probabilistic Models Decision Trees: ID3/C4.5 concepts, Gini index, entropy, pruning. Naïve Bayes Classifier: Assumptions, likelihood, posterior computation. Random Forest Bayes' Theorem: Statistical foundation with examples. Bayesian Networks: Structure, conditional dependencies, inference.	7
5	Distance-Based Learning, Clustering & MLOps Distance Metrics: Euclidean, Manhattan, Curse of dimensionality, KNN for classification and regression. Clustering: K-means, hierarchical clustering; evaluation using silhouette score. Association Rule Mining: Apriori algorithm; rule evaluation using support, confidence, lift. Overview of MLOps and its importance in productionizing ML models. Machine Learning lifecycle, Model evaluation, versioning, Inference pipelines: batch inference vs real-time inference, Model performance monitoring, Model retraining and update strategies.	9
Total Hours		39
Text Books		
1. Ethem Alpaydin, Introduction to Machine Learning, PHI Learning, 2nd Edition, 2013. ISBN: 9788120347757 2. Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012. ISBN: 9781107422223 3. Hastie, Tibshirani, Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2nd Edition, 2012. ISBN: 9780387848570 4. Tom M. Mitchell, Machine Learning, McGraw-Hill, First Edition, 1997. ISBN: 9780070428072		
Reference Books		

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 1st Edition, 2006. ISBN: 9780387310732
2. Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, 3rd Edition, 2011. ISBN: 9780123748560
3. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012. ISBN: 9780262018029
4. Parag Kulkarni, Reinforcement and Systematic Machine Learning for Decision Making, Wiley-IEEE Press, July 2012. ISBN: 9781118306352
5. Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014. ISBN: 9781107057135
6. Jacek M. Zurada, Introduction to Artificial Neural Systems, PWS Publishing, 2002. ISBN: 9780534954604

Online References

1. NPTEL Course: Prof. Balaraman Ravindran, IIT Madras, "Introduction to Machine Learning" ([Link](#))
2. NPTEL Course: Prof. Sudeshna Sarkar, IIT Kharagpur, "Introduction to Machine Learning" ([Link](#))
3. NPTEL Course: Prof. Sriram Ganapathy, Prof. Ashwini Kodipalli, Prof. Baishali Garai, IISc Bangalore RV University, "Foundations of Deep Learning: Concepts and Applications" ([Link](#))



Third Year B. Tech Information Technology														
Semester-V														
Course Code: IT24PCC302										Course Name: DevOps Engineering				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisite: Software Development Life Cycle, Cloud computing, Programming Fundamentals Operating Systems, Computer Networks														
Course Objectives: • To introduce the fundamentals of DevOps, its lifecycle, culture, and benefits. • To teach version control, CI/CD, and build automation concepts. • To explain deployment strategies, containerization, and orchestration. • To demonstrate configuration management, Infrastructure as Code (IaC), and cloud integration. • To familiarize students with monitoring, logging, and DevSecOps practices for secure software delivery.														
Course Outcomes: After learning the course, the students will be able to: CO1: Understand DevOps fundamentals, culture, roles, and version control with Git/GitHub. CO2: Apply Continuous Integration and build automation concepts using CI/CD principles. CO3: Evaluate deployment strategies and containerization using Docker. CO4: Analyze Kubernetes orchestration and configuration management with Ansible. CO5: Demonstrate cloud deployment, monitoring, logging, and basic DevSecOps practices.														
Unit	Contents													Hrs.
1	DevOps Fundamentals and Source Code Management Introduction to DevOps: Definition, Evolution and Goals of DevOps, DevOps Lifecycle Phases, Benefits of DevOps for Development, Quality Assurance and Business.DevOps Culture: CALMS Model, DevOps Roles and Responsibilities, DevOps Team Collaboration.Version Control Systems: Need for Version Control, Types of Version Control Systems – Centralized and Distributed VCS. Git Essentials: Repository Initialization, Cloning Repositories, Staging and Committing Changes, Push and Pull Operations. Branching Strategies: Branching and Merging Concepts, Git Flow, GitHub Flow, Merge Conflicts and Conflict Resolution. Collaboration using GitHub: Repositories, Issues, Forking, Pull Requests, Code Review Process. GitHub Actions: Introduction to Workflow Automation, Continuous Integration using GitHub Actions. DevOps													8

	Performance Metrics: Introduction to DevOps Metrics, DORA Metrics.	
2	Continuous Integration and Build Automation What is CI/CD, Benefits and principles of CI/CD, CI vs CD vs CD (Delivery vs Deployment), CI/CD lifecycle Continuous Integration: Concept, Need and Benefits of CI, CI Workflow. Build Management Concepts: Overview of Build Automation, Build Lifecycle. Build Tools: Overview of Maven and Gradle Concepts. Jenkins: Introduction and Jenkins Architecture, Jenkins Jobs: Freestyle Jobs and Pipeline Jobs . Jenkins Pipeline: Declarative and Scripted Pipelines (Conceptual), Jenkinsfile Structure, Integration of Version Control with CI Tools. CI Best Practices and Challenges.	8
3	Continuous Deployment and Containerization Continuous Deployment and Continuous Delivery: Concepts, Differences and Benefits. Deployment Strategies: Blue-Green Deployment, Canary Deployment, Rolling Deployment, Virtualization vs Containerization. Docker: Introduction and Docker Architecture, Docker Concepts: Images, Containers, Volumes and Networks, Dockerfile: Structure and Purpose. Container Image Repositories: Docker Hub Overview, Introduction to Docker Compose Concepts.	8
4	Container Orchestration and Configuration Management Container Orchestration: Need and Challenges, Kubernetes: Introduction and Features, Kubernetes Architecture, Kubernetes Objects, Kubernetes Networking and Scaling Concepts. Configuration Management: Introduction and Importance. Infrastructure as Code (IaC): Concepts and Benefits. Ansible: Introduction, Architecture, Inventory, Playbooks and Modules .	8
5	Cloud, Monitoring, Logging and DevSecOps Cloud Computing Fundamentals, DevOps and Cloud Integration., Cloud Platforms Overview: AWS, Azure and GCP, Cloud-based Application Deployment Concepts. Monitoring and Logging: Importance in DevOps. Monitoring Tools: Prometheus and Grafana Overview. Logging Tools: ELK Stack Overview. DevSecOps: Introduction, Security in DevOps Lifecycle and Basic Security Practices.	7
Total Hours		39
Text Books		

1. "DevOps for Developers", Michael Hüttermann, 1st Edition, Apress, 2012
2. "Practical DevOps", Joakim Verona, 1st Edition, Packt Publishing, 2020
3. "Learning Continuous Integration with Jenkins", Nikhil Pathania, 1st Edition, Packt Publishing, 2015
4. "Docker in Practice", Ian Miell, Aidan Hobson Sayers, 1st Edition, Manning Publications, 2016
5. "Kubernetes in Action", Marko Luksa, 2nd Edition, Manning Publications, 2018
6. "Cloud Computing: Concepts, Technology & Architecture", Thomas Erl, Ricardo Puttini, Zaigham Mahmood, 1st Edition, Prentice Hall, 2013.

Reference Books

1. "DevOps: A Software Architect's Perspective", Len Bass, Ingo Weber, Liming Zhu, 1st Edition, Addison-Wesley, 2015
2. "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", Gene Kim, Patrick Debois, John Willis, Jez Humble, 1st Edition, IT Revolution Press, 2016
3. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Jez Humble, David Farley, 1st Edition, Addison-Wesley, 2010
4. "Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations", Nicole Forsgren, Jez Humble, Gene Kim, 1st Edition, IT Revolution Press, 2018
5. "Docker Deep Dive", Nigel Poulton, 1st Edition, Leanpub, 2016

Online References

1. NPTEL Course: Mr. Ilangchezhiyan, Encora, "CICD – DevOps Automation and DevSecOps – Automation"([Link](#))
2. NPTEL Course: Mr. Ilangchezhiyan, Encora, "CAzure DevOps and Micro Services & Azure Kubernetes – Deployment Models"([Link](#))

Third Year B. Tech Information Technology														
Semester-V														
Course Code: IT24PCC303							Course Name: Theory of Computation							
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisites: Python programming concepts, statistics and probability														
Course Objectives: - To know the applicability of the model of computation to different problems. - To understand in detail the relationship among formal languages, formal grammars and automata. - To learn the design of Finite Automata, Pushdown Automata and Turing Machine for processing of formal languages. - To study the theory of computability and complexity for algorithm design.														
Course Outcomes: After learning the course, the students will be able to: C01: Analyze formal languages; design automata with DFA/NFA conversion and minimization. C02: Construct regular expressions and grammars; apply pumping lemma, equivalence testing. C03: Analyze CFG ambiguity, simplify grammars, apply CNF/GNF, construct PDA models. C04: Design Turing machines, classify languages, distinguish decidable and undecidable problems. C05: Classify P/NP problems, apply reductions, and identify NP-complete problems including.														
Unit	Contents										Hrs.			
1	Foundations & Finite Automata Basic Concepts: Symbols, Strings, Languages, Formal Languages FA / FSM: Formal definition & notations, State transition diagram, Transition table Construction of: DFA, NFA, NFA with ϵ -moves Conversions: NFA with ϵ -moves $\rightarrow$ NFA, NFA $\rightarrow$ DFA, NFA with ϵ -moves $\rightarrow$ DFA Minimization of FA, Equivalence of FA, Applications of FA										9			
2	Regular Expressions & Regular Grammar Regular Expressions: Definition, Operators, Identities & Identities of RE Equivalence: RE $\leftrightarrow$ Regular Languages, Equivalence of two REs Conversions: RE $\rightarrow$ FA (direct method), FA $\rightarrow$ RE (Arden's theorem) Pumping Lemma for RL										8			

	Closure properties of RL, Applications of RE Grammar: Introduction, Chomsky Hierarchy, Formal definition of Regular Grammar Conversions: $LRG \leftrightarrow RLG$, $RG \rightarrow FA$, $FA \rightarrow RG$	
3	CFG, PDA & Post Machines Context Free Grammar: Definition, Derivation tree, Sentential forms Leftmost & Rightmost derivations Ambiguity: Ambiguous & Unambiguous grammar, Context Free Languages Grammar simplification, Normal forms: CNF, GNF Closure properties of CFL Pushdown Automata: Definition, Transition diagram, Transition table Instantaneous Description (ID), DPDA vs NPDA Acceptance Equivalence: Final State $\leftrightarrow$ Empty Stack Conversions: $CFG \rightarrow PDA$, $PDA \rightarrow CFG$ Post Machine: Definition & Construction	8
4	Turing Machines & Computability Turing Machine: Formal definition, Designing TMs, Examples Variants: Deterministic TM, Nondeterministic TM, Multi-tape TM Universal Turing Machine, Church-Turing thesis Recursive vs Recursively Enumerable languages Halting Problem Post Correspondence Problem Decidability & Undecidability: Decidable problems for RL & CFL Undecidable problems	8
5	Complexity & NP-Completeness Time and Space Complexity Classes, NP, NP-Complete, NP-Hard problem basics Reduction as used in Complexity Theory, Brief overview of computational limits	6
Total Hours		39
List of Activities		
Note: Submission of these activities is mandatory for evaluation. - Design an ε-NFA for a given regular language and obtain its equivalent minimized DFA, proving equivalence at each conversion stage. - Construct equivalent Regular Expressions, Finite Automata, and Regular Grammars for a given regular language and justify regularity using the Pumping Lemma.		

3. Develop a Context-Free Grammar for a given language, remove ambiguity, convert it to CNF and GNF, and construct an equivalent Pushdown Automaton.
4. Design appropriate Turing Machines for a given problem and analyze its computability by classifying the language as recursive or recursively enumerable.
5. Classify a given decision problem into appropriate complexity classes (P, NP, NP-Hard, NP-Complete) using polynomial-time reductions.

Text Books

1. John C. Martin, "Introduction to Language and Theory of Computation", TMH, 3rd Edition, ISBN: 978-0070660489.
2. Vivek Kulkarni, "Theory of Computation", Oxford University Press, ISBN13 : 978-0198084587.

Reference Books

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory Languages and Computation", Addison-Wesley, ISBN 0-201-44124-1.
2. K.L.P Mishra, N. Chandrasekaran, "Theory of Computer Science : Automata, Languages and Computation", Prentice Hall India, 2nd Edition.
3. Michael Sipser, "Introduction to the Theory of Computation", CENGAGE Learning, 3rd Edition ISBN-13:978-81-315-2529-6.
4. Daniel Cohen, "Introduction to Computer Theory", Wiley & Sons, ISBN 97881265133454.
5. Kavi Mahesh, "Theory of Computation: A Problem-Solving Approach", Wiley India, ISBN-1081265331106.

Online References

1. E-Book: Anil Maheshwari Michiel Smid, "Introduction to Theory of Computation" ([Link](#))
2. E-Book: Sanjeev Arora and Boaz Barak, "Computational Complexity: A Modern Approach" ([Link](#))
3. NPTEL Course: Prof. Raghunath Tewari, IIT Kanpur "Theory of Computation" ([Link](#))
4. NPTEL Course: Prof. Somenath Biswas, IIT Kanpur "Theory of Computation" ([Link](#))

Third Year B. Tech Information Technology														
Semester- V														
Course Code:IT24PEC304 A										Course Name: Project Management				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisites: Software Engineering and modelling,Programming and problem solving														
Course Objectives: - To build foundational knowledge of project management concepts and practices. - To develop the ability to assess project proposals and feasibility. - To gain skills in financial planning, cost control, and project evaluation. - To understand planning, monitoring, and coordination using project management tools. - To enhance skills in effective resource allocation and scheduling.														
Course Outcomes: After completing the course, the students will be able to: C01: Understand basic project management terms, objectives, and its importance across industries. C02: Evaluate project needs and feasibility to determine viability. C03: Apply cost estimation, budgeting, and profitability analysis to assess project financial viability. C04: Analyze project tools and structures to ensure timely goals and stakeholder needs. C05: Apply resource management principles to optimize scheduling and meet project deadlines.														
Unit	Contents													Hrs.
1	Fundamentals of Project Management: Introduction to Project Management: Definition and key concepts, Objectives and significance in IT and non-IT industries, Functions of Project Management, Planning, organizing, directing, monitoring, and controlling, Evolution of Project Management, Historical developments and modern practices, Classification of Projects, based on scope, scale, domain, and complexity, Introduction to stakeholder identification and management.													7
2	Project Management Systems and PLC Methodologies: Systems Approach to Project Management, System thinking and integration, Project Requirements Analysis and Conceptualization, Requirements analysis and conceptual design, Project Feasibility and Appraisal Technical, operational, and economic feasibility, Project Life Cycle phases (Initiation, Planning, Execution, Monitoring & Controlling, Closure), Project Life Cycle Models - Predictive													8

	vs. Adaptive Project Life Cycle Models, Project Contracting, Types of contracts, Procurement and Contract Management	
3	Project Feasibility and Evaluation Techniques: Developing a Project Plan, Defining scope, deliverables, timelines, and milestones, Industry and Technical Analysis, Market demand, competitor study, technology options, Financial Analysis, Cost estimation, budgeting, profitability analysis, Evaluation of Project Proposals, Payback period, Net Present Value (NPV), Internal Rate of Return (IRR), Risk Analysis and Sensitivity Analysis, Identification, assessment, and mitigation of risks, qualitative risk analysis techniques, quantitative risk analysis techniques.	8
4	Project Planning and Organizational Structure: Planning Fundamentals: Importance of planning, stages of project planning, Project Master Plan, Integrating all components (scope, schedule, resources, risk, quality), Work Breakdown Structure (WBS) Hierarchical decomposition of work, Project Planning Tools, Gantt charts, network diagrams, milestone charts, Work Packages and Activity Definition, Project Organization Structures, Functional, matrix, and projectized, Responsibility Assignment Matrix (RAM), RACI charts and team responsibility allocation	8
5	Scheduling Tools and Techniques: PERT, CPM, and Resource Management, Introduction to PERT (Program Evaluation Review Technique): Key Characteristics of PERT, Introduction to CPM (Critical Path Method): Key Characteristics of CPM, Network Analysis (Common to PERT and CPM), Steps in Network Analysis (for both PERT and CPM), Time-Cost Relationship, optimizing project duration and cost, Resource Allocation and Leveling, Managing resource constraints and conflicts	8
Total		39 Hrs.
List of Activities		
Note: Submission of these activities is mandatory for evaluation.		
Select a single problem statement and apply all below activities to the same project.		
1. Design a simple Jira workflow with issue types, transitions, assignments, comments, and labels. 2. Create a Trello board with task lists, cards, team assignments (simulated), deadlines, and labels. 3. Develop a project plan in MS Project with tasks, durations, dependencies, resources (simulated), and track delays. 4. Set up an Asana project with sections, tasks, due dates, attachments, and team conversations (simulated).		

5. Model a project in two tools (e.g., Jira & Trello) and compare features, usability, and effectiveness.
6. Conduct risk brainstorming for a project and evaluate risks using a likelihood-impact matrix.
7. Create mitigation plans for top project risks with actions, timelines, and responsibility (simulated).

Text Books

1. Project Management: A Systems Approach to Planning, Scheduling, and Controlling Author: Harold Kerzner Publisher: Wiley
2. Information Technology Project Management Author: Kathy Schwalbe Publisher: Cengage Learning
3. Project Management for Engineering, Business and Technology Author: John M. Nicholas & Herman Steyn Publisher: Routledge

Reference Books

1. Project Management: The Managerial Process Authors: Erik W. Larson & Clifford F. Gray Publisher: McGraw-Hill
2. Project Feasibility Analysis, Author: Joy Mundy (or equivalents like Prasanna Chandra for Indian context) Publisher: Tata McGraw-Hill or Wiley
3. Software Project Management, Author: Bob Hughes, Mike Cotterell, and Rajib Mall (Indian edition) Publisher: McGraw-Hill
4. Total Project Control: A Manager's Guide to Integrated Project Planning, Measuring, and Tracking Author: Stephen A. Devaux Publisher: CRC Press
5. Feasibility Studies in Construction Projects, Author: Michael D. Bennett Publisher: Routledge
6. Managing Risk in Projects, Author: David Hillson Publisher: Routledge

Third Year B. Tech Information Technology														
Semester- V														
Course Code: IT24PEC304 B										Course Name: Internet of Things				
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60		-	-	100	3	-	-	3
Prerequisites: Fundamentals of Computer Networks,Basic Programming Knowledge														
Course Objectives: - To introduce the fundamentals and characteristics of the Internet of Things - To explain IoT architectures, models, and communication protocols - To analyze sensing and actuation mechanisms in IoT systems - To develop IoT applications with cloud integration - To assess societal and industrial applications of IoT														
Course Outcomes: After completing the course, the students will be able to: CO1: Understand the basic architecture of Internet of Things based Devices & Illustrate Wireless communication systems CO2: Gain conceptual understanding in various IoT architectures CO3: Identify actuators and sensor technologies for s for sensing real world entities. CO4: Apply design methodology and cloud platforms involved in IoT CO5: Deploy an IoT application and connect to the cloud.														
Unit	Contents													Hrs.
1	Introduction to IoT :- Introduction to Internet of Things (IoT), Functional Characteristics, Recent Trends in the Adoption of IoT, Role of cloud in IoT, Societal Benefits of IoT:- Health Care, Machine to Machine (M2M), Smart Transportation, Smart Living, Smart Cities, Smart Grid. RFID, ZigBEE, Bluetooth, Internet Communication- IP Addresses - MAC Addresses , IEEE 802 Family of Protocols , I/O interfaces Software Component.													8
2	IOT Architecture: M2M high-level ETSI architecture - IETF architecture for IoT - OGC architecture - IoT reference model - Domain model - information model - functional model - communication model.													8

3	Sensing and Actuation:- Definition of Sensor, Sensor features, Resolution, Classes, Different types of sensors, Actuator, Different types of Actuators, purpose of Sensors and Actuators in IoT	7
4	IoT Application Development:- Framework for IoT Applications-Implementation of Device integration, Data acquisition and Integration, Device data storage on cloud/local server, Authentication, authorization of Devices	8
5	Cloud computation:- Evolution of Cloud Computation, Commercial clouds and their features, open source IoT platforms, cloud dashboards, Interfacing and data logging with cloud: Blync, Thing speak, platforms.	8
Total Hours		39
List of Activities		
1. Basics of Internet of Things: Sensors, Actuators, IoT architecture and Gateway. 2. IoT Networking: Connectivity technologies, Protocols and Interoperability in IoT. 3. LED blinking with Arduino Uno. 4. IoT sensors interface with Arduino (Temperature/Light sensors). 5. Capture Image with node MCU. 6. Integration of Actuators with nodeMCU (Servo motor/Relay). 7. To interface LCD and real time clock with Arduino. 8. Upload/download sensor data on cloud and server. 9. Send live sensor data (e.g., temperature) to Thingspeak or Blynk and create graphs.		
Text Books		
1. Honbo Zhou, "The Internet of Things in the Cloud: A Middleware Perspective", CRC Press, 2012. 2. Adrian McEwen, Hakin Cassimally, "Designing The Internet of Things", First Edition, Wiley, 2014. 3. Keysight Technologies, "The Internet of Things: Enabling Technologies and Solutions for Design and Test", Application Note, 2016		
Reference Books		
1. Raj Kamal , " Internet of Things: Architecture and Design", McGraw Hill. 2nd edition June 2022 2. Peter Waher, "Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3", First Edition, Packt Publishing, 2018 3. Pethuru Raj, Anupama C. Raman , " The Internet of Things Enabling Technologies, Platforms, and Use Cases", Taylor and Francis group. February 2017		
Online References		
1. NPTEL Course: Prof. Sudip Misra, IIT Kharagpur, "Introduction To Internet Of Things " (Link) 2. Coursera Course: Introduction to the Internet of Things and Embedded Systems, (Link) 3. Industry-leading solutions to simplify and scale IT for industrial networks, (Link)		

Third Year B. Tech Information Technology														
Semester-V														
Course Code : IT24PEC304 C							Course Name: Virtualization & Cloud Computing							
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Prerequisite: Basic of Computer Network, DBMS, OS														
Course Objectives: - Understand basic concepts and evolution of cloud computing. - Understand virtualization concepts, hypervisors, and virtual machines. - Gain knowledge of major cloud platforms and services such as AWS - Get basic knowledge of cloud platforms, storage, networking, and security.														
Course Outcomes: After completing the course, the students will be able to: CO1: Demonstrate Cloud Computing Concepts and Models CO2: Explore cloud computing technologies like virtualization, storage, and networking in the context of cloud computing CO3: Analyze and describe the core offerings and services Major Cloud Platforms CO4: Create and manage virtual machines, cloud databases, and storage solutions in a cloud environment CO5: Apply cloud networking concepts to design and implement Virtual Private Cloud (VPC) architectures														
Unit	Contents										Hrs.			
1	Introduction to Cloud Computing : Recent trends in computing, Cluster computing, Distributed computing ,Evolution of cloud computing, Cloud versus traditional architecture ,Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Cloud Deployment Models: Public cloud, Private cloud, Hybrid cloud, Community cloud										7			
2	Virtualization: Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM), Configuring elastic apps with auto scaling, Basics of virtualization and implementation challenges. System virtualization technologies-architectures and internals. KVM, Xen, VMware										8			

3	Cloud Services & platform : Amazon Web Services (AWS): Overview and Services (EC2, S3, Lambda, etc.)Microsoft Azure: Overview and Services (Azure VM, Blob Storage, Azure Functions)Google Cloud Platform (GCP): Overview and Services (Compute Engine, Google Cloud Storage)	8
4	Cloud Storage: Storage options in the cloud, Structured and unstructured storage in the cloud, unstructured storage using Cloud Storage, SQL managed services, Exploring Cloud SQL, Cloud Spanner as a managed service, NoSQL managed service options, Cloud Datastore, a NoSQL document store, Cloud Bigtable as a NoSQL option.	8
5	Cloud Network and Security: Introduction to networking in the cloud, defining a Virtual Private Cloud, Public and private IP address basics, Routes and firewall rules in the cloud, Multiple VPC networks, building hybrid clouds using VPNs, interconnecting, and direct peering, Different options for load balancing. Introduction to security in the cloud, the shared security model, Encryption options. Case Studies in Cloud Computing : Real-World Applications of Cloud Computing, Case Studies in Enterprises, Healthcare, Finance, and Government	8
Total Hours		39
List of Activities		
1. Classify various cloud-based services into IaaS, PaaS, and SaaS categories with real-world examples and provide justifications. 2. Set up and configure a Virtual Machine using a public cloud platform such as AWS EC2 or Azure VM and demonstrate basic connectivity and software installation. 3. Analyze and present a real-world case study on cloud adoption in a selected domain such as Healthcare, Finance, or Government. 4. Demonstrate the use of cloud storage services by uploading and managing structured and unstructured data using services like AWS S3, Cloud SQL, and Cloud Datastore. 5. Design and simulate a basic cloud network architecture using VPC, subnets, and firewall rules, and explain the shared responsibility model in cloud security.		
Text Books		
1. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, "Distributed and cloud computing from Parallel Processing to the Internet of Things", Morgan Kaufmann, Elsevier – 2012 2. Zaigham Mahmood, Ricardo Puttini, Thomas Erl , "Cloud Computing: Concepts, Technology & Architecture" Pearson 3. Barrie Sosinsky, " Cloud Computing Bible" John Wiley & Sons, 2010 4. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley,India. 5. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India		

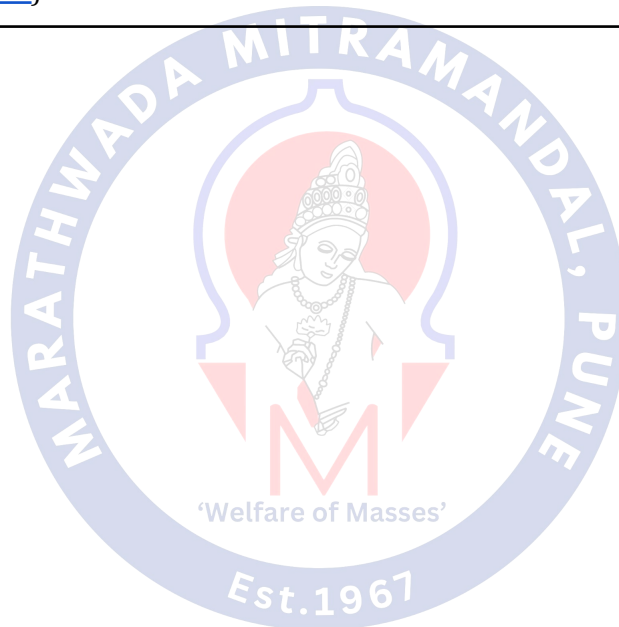
6. Gautam Shroff. "Enterprise Cloud Computing", Cambridge

Reference Books

1. Antohy T Velte, et.al, "Cloud Computing: A Practical Approach", McGraw Hill.
2. Michael Miller, "Cloud Computing", Que Publishing.
3. Tim Malhar, S.Kumaraswamy, S.Latif, "Cloud Security & Privacy", SPD,O'REILLY
4. Scott Granneman, "Google Apps", Pearson

Online References

1. NPTEL Course: Prof. Soumya Kanti Ghosh, IIT Kharagpur, Cloud Computing, ([Link](#))
2. NPTEL Course: Prof. Soumya Kanti Ghosh, IIT Kharagpur, Google Cloud Computing Foundations Course, ([Link](#))
3. Coursera: Reza Farivar el. At. , Cloud Computing Specialization, ([Link](#))
4. Aws Documentation:([Link](#))



Third Year B. Tech Information Technology														
Semester-V														
Course Code : IT24PEC304 D							Course Name: Game Design and Development							
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
3	-	-	-	-	40	60	-	-	-	100	3	-	-	3
Note: As this is a lab-weighted industry elective with no traditional end-semester theory examination, evaluation is based fully on continuous in-semester assessment (CIE) and a final term-end practical/capstone evaluation (ETE), as detailed in the Assessment Structure below.														
Prerequisite: Basic computer literacy and familiarity with using a desktop operating system are assumed.														
Course Objectives: - To introduce students with no prior exposure to the fundamental concepts of game design and game development. - To build proficiency in navigating and using the Unity game engine for 2D game development. - To develop working knowledge of C# scripting for implementing game logic, player control, and interactivity. - To enable students to apply physics-based collision systems and scoring mechanics within a game. - To cultivate the ability to design, build, playtest, and iterate on a complete, playable 2D game independently.														
Course Outcomes: After completing the course, the students will be able to: C01: Explain core game design concepts, including the core loop, mechanics, dynamics, and aesthetics (MDA framework). C02: Design and playtest a simple game on paper, and apply playtester feedback through iteration. C03: Navigate the Unity editor and apply its core building blocks (GameObjects, Components, Prefabs) to construct 2D scenes. C04: Write basic C# scripts to control player movement and implement game logic. C05: Implement physics-based collision detection and scoring systems, and build and present one complete, playable 2D game.														
Unit	Contents										Hrs.			

1	Foundations of Game Design: What Even Is a Game: Games vs. simulations vs. toys, the core loop, Mechanics, Dynamics, and Aesthetics (MDA framework, simplified). Paper Prototyping: Why prototype before code, defining rules, win/lose conditions and player goals, reading and applying playtester feedback.	8
2	Unity Orientation and Environment Setup: Navigating the Unity Editor: Scene, Hierarchy, Inspector, and Project panels. GameObjects and Components. Introduction to Prefabs (concept only). Building static 2D scenes: placing sprites, setting up a camera, and arranging a simple environment.	5
3	Scripting Basics in C#: Fundamentals of C# for game logic: variables, conditionals (if/else), and loops in a game context. Working with Transform.position and the Update() loop. Reading keyboard input and implementing keyboard-controlled player movement with screen/scene boundaries.	9
4	Physics, Collision, and Gameplay Logic: Rigidbody2D and Collider components. Differentiating colliders vs. triggers. Collision detection fundamentals and driving gameplay logic through collisions. Implementing a collectible pickup mechanic with disappearing objects and an increasing score.	6
5	Game Feel, UI, and Capstone Integration: Why feedback ('game feel'/juice) matters to player experience. Building basic in-game UI: score display, buttons, and simple sound effects. Capstone Project: integrating movement, collision, scoring, and UI into one complete playable 2D game; structured peer playtesting and feedback.	11
Total Hours		39

List of Activities

Note: Submission of these activities is mandatory for evaluation.

1. Deconstruct one published 2D game on paper, documenting its core loop, mechanics, dynamics, and win/lose conditions.
2. Design and playtest a paper/board game prototype in pairs; document playtester feedback and the resulting design iteration.
3. Construct a static Unity 2D scene using sprites, camera setup, and basic environment layout.
4. Write a C# script implementing keyboard-controlled player movement constrained within defined screen boundaries.
5. Implement a collectible pickup mechanic using Rigidbody2D and Collider/Trigger components, with an increasing score on pickup.

6. Add a live in-game UI (score display, restart button) and integrate one sound effect into the existing project.
7. Build and submit one complete, playable 2D game integrating movement, collision, scoring, and UI; participate in structured peer play testing.

Text Books

1. Introduction to Game Design, Prototyping, and Development — Author: Jeremy Gibson Bond, Publisher: Addison-Wesley / Pearson
2. The Art of Game Design: A Book of Lenses — Author: Jesse Schell — Publisher: CRC Press
3. Unity in Action: Multiplatform Game Development in C# — Author: Joseph Hocking — Publisher: Manning Publications

Reference Books

1. Game Programming Patterns — Author: Robert Nystrom — Publisher: Genever Benning
2. Rules of Play: Game Design Fundamentals — Authors: Katie Salen Tekinbaş & Eric Zimmerman — Publisher: MIT Press
3. Beginning C# Game Programming — Author: Kelvin Sung et al. — Publisher: CRC Press

Online References

1. Unity Learn — Official Unity Technologies tutorials and learning pathways for 2D game development.
2. Coursera Course: Michigan State University, Game Design and Development with Unity.
3. Coursera Course: University of Colorado System, Introduction to Game Development.
4. Unity Documentation — Official Unity Manual and Scripting API Reference.

Third Year B. Tech Information Technology														
Semester-V														
Course Code: IT24PCC306					Course Name: Data Science & Machine Learning Lab									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	PR	OR	TOTAL	L	P	T	TOTAL
-	2	-	-	-	-	-	-	50	-	50	-	1	-	1
Prerequisites: Python programming concepts, statistics and probability														
Course Objectives: - To understand data exploration, preprocessing, and statistical analysis techniques. - To apply regression methods for prediction and performance evaluation. - To implement and compare supervised machine learning classification algorithms. - To learn clustering techniques for pattern discovery in datasets. - To evaluate and interpret machine learning models using appropriate metrics and visualization tools.														
Course Outcomes: After completing the course, the students will be able to: C01: Apply data exploration, descriptive statistics, and validation techniques on datasets. C02: Build and evaluate regression models using appropriate performance metrics. . C03: Implement classification algorithms such as Logistic Regression, SVM, Decision Tree, Naïve Bayes, and KNN. C04: Apply clustering techniques to discover patterns and evaluate cluster quality. C05: Analyze and interpret machine learning model performance using visualization and evaluation metrics.														
List of Experiments														
1	Data Exploration and Descriptive Statistics Load the Iris dataset and perform exploratory data analysis by computing mean, median, standard deviation, skewness, and kurtosis for numeric features. Visualize the distribution using histograms and boxplots. Dataset Source: sklearn.datasets.load_iris() or UCI ML Repository – Iris													2
2	Sampling and Validation Techniques Use the Titanic dataset to split data into training and test sets using Hold-Out, K-Fold, and Leave-One-Out Cross-Validation (LOOCV). Compare performance variation across these strategies using a basic classifier.													2

	Dataset Source: seaborn.load_dataset('titanic') or Kaggle Titanic Dataset	
3	Simple Linear Regression and Error Metrics Implement simple linear regression on the Student Scores dataset to predict marks based on hours studied. Evaluate performance using MSE, MAE, and R^2 . Dataset Source: Student Scores CSV (https://shorturl.at/x1jjw)	2
4	Multivariate and Polynomial Regression Use the California Housing dataset to perform multivariate linear regression and polynomial regression. Dataset Source: sklearn.datasets.fetch_california_housing()	2
5	Logistic Regression and ROC-AUC Use the Titanic dataset to build a logistic regression model to predict survival. Evaluate using confusion matrix, precision, recall, F1-score, and plot the ROC curve. Dataset Source: seaborn.load_dataset('titanic') or Kaggle Titanic Dataset	2
6	Support Vector Machines (SVM) Apply SVM on the Iris dataset to classify species. Dataset Source: sklearn.datasets.load_iris()	4
7	Decision Tree and Pruning Techniques Build a decision tree on the Iris dataset. Use entropy and Gini index as criteria. Apply pruning and compare performance before and after pruning. Dataset Source: sklearn.datasets.load_iris()	2
8	Naïve Bayes Classification Use the SMS Spam dataset to build a Naïve Bayes classifier. Preprocess the text data, calculate prior and likelihood probabilities, and evaluate accuracy. Dataset Source: UCI SMS Spam Dataset	4
9	K-Nearest Neighbors (KNN) with Distance Metrics Use the Iris dataset to implement KNN classification with Euclidean and Manhattan distance metrics. Compare accuracy and decision boundaries for different values of K. Dataset Source: sklearn.datasets.load_iris()	4
10	Apply K-Means clustering on the Wine dataset. Determine optimal number of clusters using the elbow method. Evaluate cluster quality using silhouette score and interpret findings. Dataset Source: sklearn.datasets.load_wine() or UCI Wine Dataset	2
Total		26
Text Books		

1. Ethem Alpaydin, *Introduction to Machine Learning*, PHI Learning Pvt. Ltd., 2nd Edition, 2013. ISBN: 9788120347757.
2. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 1st Edition, 1997. ISBN: 9780070428072.
3. Andreas C. Müller and Sarah Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, 1st Edition, 2016. ISBN: 9781449369415.
4. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 1st Edition, 2006. ISBN: 9780387310732.

Reference Books

1. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 1st Edition, 2012. ISBN: 9780262018029.
2. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 3rd Edition, 2022. ISBN: 9781098125974.
3. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2nd Edition, 2009. ISBN: 9780387848570.
4. Sebastian Raschka and Vahid Mirjalili, *Python Machine Learning*, Packt Publishing, 3rd Edition, 2019. ISBN: 9781789955750.

Online References

1. NPTEL Course: Prof. Balaraman Ravindran, IIT Madras, "Introduction to Machine Learning" ([Link](#))
2. NPTEL Course: Prof. Sudeshna Sarkar, Indian Institute of Technology Kharagpur, *Introduction to Machine Learning*.
3. NPTEL Course: Prof. Balaraman Ravindran, Indian Institute of Technology Madras, *Introduction to Machine Learning*.
4. Coursera Course: Andrew Ng, Stanford University, *Machine Learning*.
5. Online Documentation: [Scikit-Learn Documentation](#)
6. Online Documentation: [Pandas Documentation](#)
7. Online Repository: [UCI Machine Learning Repository](#)
8. Online Platform: [Kaggle Datasets and Competitions](#)
9. Online Platform: [Google Colaboratory \(Colab\)](#)

Third Year B. Tech Information Technology														
Semester :V														
Course Code: IT24PCC307								Course Name: DevOps Engineering Lab						
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	2	-	-	-	-	-	-	50	-	50	-	1	-	1
Prerequisites: Operating systems,Fundamentals of programming (any language),Software development concepts														
Course Objectives: - To understand the principles, culture, and lifecycle of DevOps in modern software development. - To apply version control systems for effective source code management and team collaboration. - To implement continuous integration and build automation practices using CI tools. - To apply containerization concepts for efficient application deployment. - To explain monitoring, logging, and security practices in DevOps environments.														
Course Outcomes: After completing the course, the students will be able to: C01: Analyze DevOps principles, culture, lifecycle phases, and their impact on performance metrics. C02: Demonstrate the use of version control systems for source code management and collaborative development. C03: Implement continuous integration workflows and automated build processes. C04: Apply containerization and basic orchestration concepts for application deployment. C05: Apply basic monitoring, logging, and security practices in a DevOps workflow.														
Sr. No.	List of Experiments										Duration (Hrs.)			
1	Perform DevOps basics by initializing a repository, creating multiple versions of files, committing changes, and analyzing version history										2			
2	Create a local repository, connect it with a remote repository, and perform clone, push, pull, and commit operations to manage source code.										2			
3	Create feature branches, perform merge operations, and resolve merge conflicts to understand branching strategies.										4			
4	Use a version control platform to collaborate by creating repositories, raising issues, forking projects, submitting pull requests, and reviewing code.										4			

5	Design and implement a CI workflow that automatically builds or tests a project whenever code is pushed to the repository.	4
6	Create and execute an automated build job integrated with a version control system to understand build automation concepts.	4
7	Create a container configuration file, build container images, and run containers to understand application containerization.	4
8	Design and implement inventory files and automation scripts using Ansible to manage configuration tasks across multiple systems	4
Total Hours		28
Text Books		
1. Len Bass, Ingo Weber, Liming Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley, 2015 2. Gene Kim, Jez Humble, Patrick Debois, John Willis, The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, IT Revolution Press, 2016. 3. Scott Chacon, Ben Straub, Pro Git, Apress, 2014. 4. Jez Humble, David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley, 2010.		
Reference Books		
1. DevOps: A Software Architect's Perspective, Len Bass, Ingo Weber, Liming Zhu, 1st Edition, Addison-Wesley, 2015 2. The DevOps Handbook, Gene Kim, Jez Humble, Patrick Debois, John Willis, 1st Edition, IT Revolution Press, 2016 3. Pro Git, Scott Chacon, Ben Straub, 2nd Edition, Apress, 2014 4. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Jez Humble, David Farley, 1st Edition, Addison-Wesley, 2010 5. Docker Deep Dive, Nigel Poulton, 1st Edition, Packt Publishing, 2017		
Online References		
1. Introduction to DevOps, Great Learning Academy (Link) 2. Continuous Integration & Continuous Deployment with Jenkins (Link) 3. Master DevOps: CI/CD, Automation & Monitoring Specialization, (Link)		

Third Year B.Tech Information Technology														
Semester-V														
Course Code: IT24PCC308					Course Name: : Engineering Design and Innovation-I									
Teaching Scheme (Hours/Week)					Examination Scheme						Credits			
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	TOTAL
-	4	-	-	-	-	-	50	-	-	50	-	2	-	2
Prerequisite: Basic Programming Knowledge (C / C++), Fundamentals of Object-Oriented Programming, Basic Data Structures (Array, Stack, Queue, Linked List), Data Base Management Systems.														
Course Objectives: - To enable students to integrate and apply concepts from multiple foundational engineering courses to solve real-world problems. - To develop system-level thinking by transforming problem statements into structured requirements, design, and implementable solutions. - To provide hands-on experience in using relevant engineering tools, techniques, and technologies for building and testing functional prototypes or systems. - To enhance teamwork, communication, documentation, and project management skills through collaborative project execution and presentation.														
Course Outcomes: After learning the course, the students will be able to: C01: Identify and analyze an engineering problem by integrating knowledge from multiple foundational courses. C02: Design and develop a feasible engineering solution using appropriate tools, technologies, and methodologies. C03: Test, validate, and refine the developed solution while applying systematic problem-solving and quality evaluation techniques. C04: Demonstrate effective teamwork, project management, technical documentation, and professional communication through project presentation and reporting.														
Engineering Design and Innovation-I exclusively devoted to a Mini project-based learning component as per the guidelines of APNI Leap- Mini Project Foundation Integration Course, to be pursued continuously throughout the semester. Students are required to initiate the Mini- project work at the very beginning of the semester.														
Team Formation and Mentoring: The course shall be conducted through mentor-guided, team-based learning. Students shall work in teams of 3–4 members under the supervision of a faculty mentor to identify, design, develop, and validate an innovative software solution. The mentor shall monitor progress through periodic reviews and provide technical guidance throughout the semester.A supervisor/mentor teacher assigned to individual groups														
Problem Identification and Project Selection:														

Students shall undertake a semester-long mini project based. The selected problem should address a real-world engineering, industrial, institutional, or societal need and integrate concepts from fundamental Information Technology courses. The project shall follow the engineering design process comprising problem identification, requirement analysis, system design, prototype development, testing, validation, and documentation. Modern software development practices shall be employed for implementation. Students are required to select an appropriate programming language, preferably Java, or another suitable language based on the project requirements, for the implementation of their project.

Assessment/Evaluation Methodology:

The Mini Project Coordinator is committed to assessing and evaluating both student performance and program effectiveness. Progress is monitored regularly on a weekly basis. Weekly review of the work is necessary.

During the process of monitoring and continuous assessment AND evaluation the individual and team performance is to be measured. Mini Project is monitored and continuous assessment is done by supervisor/mentor

and authorities. The institution/department should support students in this regard through guidance/orientation programs and the provision of appropriate resources and services.

Supervisor/mentor and Students must actively participate in assessment and evaluation processes.

The group may demonstrate their knowledge and skills by developing a public product and/or report and/or presentation.

- Individual assessment for each student (Understanding individual capacity, role and involvement in the project)
- Group assessment (roles defined, distribution of work, intra-team communication and togetherness)
- Documentation and presentation

Mini-Project Themes with Associated Fundamental Courses'

Sr. No.	Mini-Project Problem Statement	Fundamental Courses Likely to be Applied
1	Smart Inventory Management System: Design and develop a system to automate inventory tracking, supplier management, stock transactions, and report generation with secure user management and database-driven operations.	Programming Fundamentals, Object-Oriented Programming (OOP), Data Structures, Database Management Systems (DBMS), Software Engineering, Web Technologies
2	Student Information Management System: Design and develop a system to manage student records, attendance, academic performance, and course enrollment with secure access and reporting.	Programming Fundamentals, OOP, DBMS, Software Engineering, Data Structures, Web Technologies
3	Employee Payroll Management System: Design and develop a system to automate employee information management, payroll processing, leave management, and salary report generation.	Programming Fundamentals, OOP, DBMS, Software Engineering, Data Structure
4	Library Management System: Design and develop a system to automate book cataloging, issue/return transactions, member management, and fine calculation.	Programming Fundamentals, OOP, DBMS, Data Structures, Software Engineering
5	Hospital Management System: Design and develop a system to manage patient records, appointments, billing, and medical reports with secure and scalable architecture.	Programming Fundamentals, OOP, DBMS, Software Engineering, Web Technologies
6	Online Examination System: Design and develop a secure platform for examination scheduling, question management, automatic evaluation, and result generation.	Programming Fundamentals, OOP, DBMS, Software Engineering, Computer Networks, Web Technologies
7	Smart Campus Management System: Design and develop a system to manage campus resources, student services, event notifications, and administrative activities.	Programming Fundamentals, OOP, DBMS, Software Engineering, Computer Networks, Web Technologies
8	E-Commerce Order Management System: Design and develop a system to manage products, customers, orders, inventory, and payments with scalable architecture	Programming Fundamentals, OOP, DBMS, Data Structures, Software Engineering, Web Technologies
9	Vehicle Rental Management System: Design and develop a system to automate vehicle booking, rental transactions, customer management, and payment processing.	Programming Fundamentals, OOP, DBMS, Data Structures, Software Engineering

10	Complaint and Service Request Management System: Design and develop a system for complaint registration, tracking, resolution, notifications, and reporting.	Programming Fundamentals, OOP, DBMS, Software Engineering, Web Technologies
11	Web-Based Inventory Management System: Design and develop a web-based inventory application with authentication, session management, CRUD operations, and reporting	OOP, DBMS, Web Technologies, Software Engineering, Computer Networks
12	Reactive Enterprise Application: Design and develop a high-performance application supporting asynchronous processing, concurrent users, and responsive services	OOP, Operating Systems, Computer Networks, Software Engineering, Distributed Systems
13	Cloud-Ready Enterprise Application: Design and develop a cloud-ready application with caching, logging, monitoring, and performance optimization.	OOP, Operating Systems, Cloud Computing, Software Engineering, Computer Networks
14	Secure REST-Based Management System: Design and develop a secure service-oriented application supporting authenticated and authorized access to enterprise resources.	OOP, Computer Networks, Cybersecurity, Web Technologies, DBMS, Software Engineering
15	Smart Notification and Resource Management System: Design and develop a system using modular architecture and event-driven communication for resource management and automated notifications	OOP, Software Engineering, Design Patterns, Data Structures, Computer Networks

Text Books

1. Y. Daniel Liang, "Introduction to Java Programming and Data Structures, Comprehensive Version", Pearson. [\(Link\)](#)
2. Herbert Schildt, "Java: The Complete Reference", McGraw Hill Education. [\(Link\)](#)
3. Software Engineering: A Practitioner's Approach, Roger S. Pressman and Bruce R. Maxim, McGraw Hill Education.
4. Software Engineering, Ian Sommerville, Pearson Education.
5. Object-Oriented Analysis and Design with Applications, Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Pearson Education.
6. Database System Concepts, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill Education.
7. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Addison-Wesley.

Reference Books

1. Joshua Bloch, "Effective Java", Addison-Wesley.
2. Norman, D. *The Design of Everyday Things*.
3. Kathy Sierra & Bert Bates, "Head First Java", O'Reilly Media.
4. Murach's, "Murach's Java Servlets and JSP", Mike Urban et al.
5. Cay S. Horstmann, *Core Java Volume I – Fundamentals*, Pearson Education.
6. Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Modeling Language User Guide*, Addison-Wesley.
7. Cay S. Horstmann, *Core Java Volume II – Advanced Features*, Pearson Education.
8. Eric Freeman, Elisabeth Robson, *Head First Design Patterns*, O'Reilly Media.

Online References

1. Coursera Course: "Object Oriented Java Programming: Data Structures and Beyond" – University of California San Diego ([Link](#))
2. Coursera Course: "Data Structures & Backend with Java" – Board Infinity ([Link](#))
3. Udemy Course: "Complete Java Programming and Data Structures & Algorithms" – Sankara Rao P ([Link](#))
4. GitHub Skills: "Introduction to GitHub ([Link: https://skills.github.com](https://skills.github.com))



Third Year B.Tech Information Technology																
Semester-V																
Course Code: IT24AEC311 Course Name: Analytical Reasoning and Universal Human Values-I																
Teaching Scheme (Hours/Week)					Examination Scheme						Credits					
L	P	T	OL	ODL	CIE	ETE	TW	OR	PR	TOTAL	L	P	T	OL	ODL	TOTAL
-	-	1	-	-	-	-	25	-	-	-	-	-	1	-	-	1
Prerequisite: Basic Mathematical Concepts																
Course Objectives: - To develop fast and accurate mental calculation skills using Vedic Mathematics techniques for arithmetic operations. - To strengthen students' understanding of number systems, divisibility rules, and numerical properties for efficient problem solving. - To enhance competency in percentages, profit-loss, discounts, ratios, and partnerships for real-life and competitive exam applications. - To build logical reasoning abilities through linear, circular, and complex puzzle-based arrangements. - To improve analytical thinking by applying concepts of syllogisms, blood relations, and direction sense. - To prepare students for aptitude-based assessments and competitive examinations by integrating speed, accuracy, and logical reasoning skills.																
Course Outcomes: After learning the course, the students will be able to: C01: Perform rapid mental calculations involving addition, subtraction, multiplication, squaring, square roots, and cube roots with high accuracy. C02: Analyze and solve number system problems involving divisibility, unit digits, factorials, HCF-LCM, and remainders. C03: Apply percentage, profit-loss, discount, ratio, and partnership concepts to solve quantitative aptitude problems efficiently. C04: Solve seating arrangement and puzzle-based problems, including linear, circular, matrix, and scheduling puzzles, using logical reasoning. C05: Evaluate syllogism problems and blood relation cases using Venn diagrams, family trees, and logical inference. C06: Demonstrate improved speed, accuracy, and confidence in quantitative and reasoning sections of placement and competitive examinations.																
Unit	Contents - Foundations & Numerical Proficiency										Duration (Hrs.)					

1	Vedic Math & Mental Calculation • Speed Addition & Subtraction (Left-to-Right method) • Base Method Multiplication (Numbers near 10, 100, 1000) • Squaring Techniques (Base 50/100 and numbers ending in 5) • Cube Roots and Square Roots of perfect numbers	2
2	Number Systems • Classification (Rational, Irrational, Prime, Composite) • Divisibility Rules (2 to 19) • Unit Digit and Cyclicity • Factorial theory and Number of Zeroes • HCF & LCM (Models: Remainder cases) • Remainder Theorem basics	2
3	Percentages • Percentage Increase/Decrease and Multiplying Factors • Successive Percentage Change Formula • Product Constancy Rule	2
4	Profit, Loss & Discount • CP, SP, and Profit/Loss Percentage • Relation between Markup, Discount, and Profit • Successive Discounts and Single Equivalent Discount	2
5	Ratio, Proportion & Partnership • Ratio Scaling and Merging (A:B and B:C to A:B:C) • Mean, Third, and Fourth Proportional • Coin-based problems • Partnership: Profit sharing based on (Capital $\times$ Time) • Problems on Ages	2
6	Linear Arrangements • Single Row: All facing North/South • Dual Row: Facing each other • Directional cues (Left of, Second to the right, Immediate neighbours) • Ranking and Ordering (Tallest/Shortest, Top/Bottom)	2
7	Circular & Polynomial Arrangements • Circular: Facing Centre vs. Facing Outwards • Inward/Outward Mixed configurations • Square and Rectangular Table arrangements	2

	Relationship-based seating (Combining Seating + Blood Relations)																
8	Complex Puzzles - Grid/Matrix Puzzles (3 or 4 variables: Name, City, Profession, Color) - Scheduling Puzzles (Days of the week, Months, Years) - Floor-based Puzzles - Box-based Puzzles (Stacking logic)	2															
9	Syllogisms - Standard Propositions (All, Some, No, Some Not) - Venn Diagram Method "Only a Few" and "Few" cases - Possibility-based conclusions - Complementary pairs (Either-Or cases)	2															
10	Blood Relations & Direction Sense - Family Tree Construction (Standard symbols) - Coded Blood Relations - Direction Compass and Degree Turns	2															
Total Hours		20															
Group B																	
Self and Body Harmony: Identify human aspirations and analyze harmony between self and body																	
Family Relationships: Analyze trust and respect in family relationships with examples																	
Social Harmony Explain social harmony and the concept of universal human order																	
Reference Books																	
<table border="1"> <tbody> <tr> <td>Quant Basics</td><td>R.S. Aggarwal</td><td>Concept Clarity</td></tr> <tr> <td>Speed Math</td><td>M. Tyra</td><td>Time Management</td></tr> <tr> <td>Logic/Puzzles</td><td>K. Kundan / R.S. Aggarwal</td><td>Exam-specific Patterns</td></tr> <tr> <td>Critical Thinking</td><td>M.K. Pandey</td><td>Logical Derivations</td></tr> <tr> <td>DI / Advanced</td><td>Arun Sharma</td><td>Complex Data & CAT</td></tr> </tbody> </table>			Quant Basics	R.S. Aggarwal	Concept Clarity	Speed Math	M. Tyra	Time Management	Logic/Puzzles	K. Kundan / R.S. Aggarwal	Exam-specific Patterns	Critical Thinking	M.K. Pandey	Logical Derivations	DI / Advanced	Arun Sharma	Complex Data & CAT
Quant Basics	R.S. Aggarwal	Concept Clarity															
Speed Math	M. Tyra	Time Management															
Logic/Puzzles	K. Kundan / R.S. Aggarwal	Exam-specific Patterns															
Critical Thinking	M.K. Pandey	Logical Derivations															
DI / Advanced	Arun Sharma	Complex Data & CAT															
Assessment																	
Assessment will be done by the Course Teacher. MCQ Test can be conducted based on the syllabus.																	