Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. Computer Engineering (Software Engineering)

With Effect from Academic Year 2026-27

Prepared By: Board of Studies in CE(SE)

Approved By: Academic Board, Vishwakarma Institute of Technology, Pune

Chairman – BoS **Chairman – Academic Board**

Index

Sr. No.	Particular	Page No
1	Vision Mission of Institute	III
2	Vision Mission of Department	III
3	Board of Studies	IV
4	Program Educational Objectives	V
5	List of Program Outcomes	VI
6	List of Program Specific Outcomes	VII
7	NEP Nomenclatures	VIII
8	TY Structure	X
9	Third Year Module V Content	1

College Vision and Mission

Vision:

To be a globally acclaimed Institute in Technical Education and Research for holistic socio economic development

Mission:

- To ensure that 100% of students are employable and employed in industry, higher studies, entrepreneurship, civil or defence services, government jobs, and other areas like sports and arts.
- To strengthen Academic Practices in Curriculum, Pedagogy, Assessment and Faculty Competence.
- To Promote Research Culture among Students and Faculty through Projects and Consultancy
- To make students Socially Responsible Citizens

Department Vision and Mission

Vision:

Empowering Industry through Comprehensive Software Engineering Services to Achieve Excellence.

Mission:

- To produce industry-ready software engineering graduate with a blend of technical expertise and ethical responsibility
- To provide software engineering students with the utmost quality in developing technical, social, innovative, and entrepreneurial skills
- To drive technological innovation and shape the future of software engineering

Board of Studies

Sr. No.	Name	Affiliation in Committee	Designation
1.	Prof. (Dr.) Sunil Sangve	Head of Department (Chairman)	Professor and Head
2.	Dr. (Ms.) Leena Deshpande	Entire Faculty of Each Specialization Representative	Associate Professor
3.	Dr. (Ms.) Varsha Jadhav	Entire Faculty of Each Specialization Representative	Assistant Professor
4.	Mr. Shailesh Thaware	Entire Faculty of Each Specialization Representative	Assistant Professor
5.	Mr. Vilas Ghonge	Entire Faculty of Each Specialization Representative	Assistant Professor
6.	Prof. (Dr.) Sunil Wankhede	Subject Expert from Outside University nominated by Academic Council	Professor, Rajiv Gandhi institute of Technology, Mumbai
7.	Prof. (Dr.) Shashank Joshi	Subject Expert from Outside University nominated by Academic Council	Professor, Bharati Vidyapeeth Deemed University, Pune
8.	Prof. (Dr.) Nilesh Uke	VC Nominee	Principal, Indira College of Engineering, Pune
9.	Mr. Dinesh Kothawade	Industry Representative	TCS, Delivery Partner
10.	Mr. Himanshu Tyagi	Meritorious Alumnus Nominated by Director	Sr. Software Developer LinkedIn

Programme Education Objectives (PEO)

PEO	PEO Focus	PEO Statement
1	Core Competence	Demonstrate strong foundational knowledge and technical skills in software engineering principles and practices to solve real-world problems.
2	Application & Innovation	Apply engineering principles, contemporary software development methodologies, and modern computational tools to design and develop efficient, innovative, and scalable software systems that address complex real-world problems and solutions
3	Professionalism & Ethics	Exhibit professionalism, ethical behavior, effective communication, and teamwork in diverse and multidisciplinary environments.
4	Lifelong Learning & Entrepreneurship	Engage in lifelong learning, pursue higher education, research, or entrepreneurship in emerging areas of software engineering and allied fields.

Program Outcomes (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSO)

- PSO 1 – Software Design & Development
  Analyze, design, and implement reliable, efficient, and scalable software solutions by applying software engineering principles, modern programming tools, and innovative methodologies to solve real-world problems.
- PSO 2 – Professional Growth & Innovation
  Demonstrate professionalism, ethics, effective communication, and teamwork, while engaging in lifelong learning, research, Industry and entrepreneurial activities to contribute to emerging areas.

Course Name Nomenclature as per NEP

BSC: Basic Science Course

ESC: Engineering Science Course

PCC: Program Core Course

PEC: Program Elective Course

ELC: Experiential Learning Course

MD: Multi-Disciplinary

MDOE: Multi-Disciplinary Open Elective

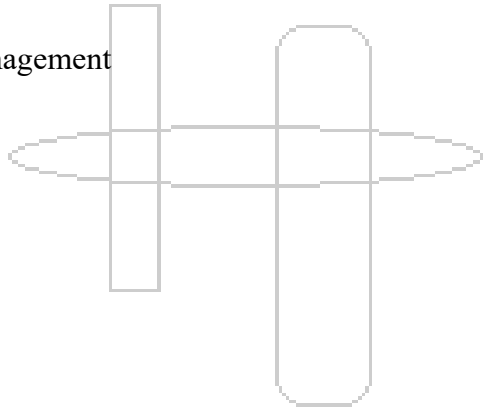
CC: Co-curricular Course

HSSM: Humanities Social Science and Management

IKS: Indian Knowledge System

FP: Field Project

INT: Internship



Nomenclature for Teaching and Examination Assessment Scheme

Sr No.	Category	Head of Teaching/ Assessment	Abbreviation used
1	Teaching	Theory	Th
2	Teaching	Laboratory	Lab
3	Teaching	Tutorial	Tut
4	Teaching	Open Elective	OE
5	Teaching	Multi-Disciplinary	MM
6	Teaching	Software Engineering	SE
7	Assessment	Laboratory Continuous Assessment	CA
8	Assessment	Mid Semester Assessment	MSA
9	Assessment	End Semester Assessment	ESA
10	Assessment	Home Assignment	HA
11	Assessment	Course Project	CP
12	Assessment	Group Discussion	GD
13	Assessment	PowerPoint Presentation	PPT
14	Assessment	Practical	PR
15	Assessment	Mid Semester Examination	MSE
16	Assessment	End Semester Examination	ESE
17	Assessment	Written Examination	W
18	Assessment	Online Examination	O
19	Assessment	Review	R
20	Assessment	Comprehensive Viva Voce	CVV
21	Assessment	Laboratory	LAB

Title: Course Structure **FF No. 653-A**

Branch: CESE **Year:** TY **Academic Year:** 2026-27 **Semester:** I **Module:** V **Pattern:** 2024

Level 5.5																						
Course Code	NEP Course Category	Course Name	Teaching Scheme (Hrs./week)			Examination Scheme and Marks													Credits			
			Theory	Tutorial	Practical	CVV	CP	LAB/ TUT	GD/PPT	HA	MSE (W)	MSE (R)	MSE (O)	ESE(R)	ESE (W)	ESE (O)	PRACT+ CVV	ESE-Direct Evaluation	Theory	Tutorial	Practical	Total
Semester I																						
SE3201	PCC	ARTIFICIAL INTELLIGENCE IN SOFTWARE ENGINEERING	3	-	2	-	-	-	-	-	-	-	-	-	40	-	60	-	3	-	1	4
SE3202	PCC	AGILE METHODOLOGY	2	-	2	20	30	10	-	-	-	-	-	-	40	-	-	-	2	-	1	3
SE3203	PEC1	PROGRAM ELECTIVE I	3	-	2	20	40	-	20	20	-	-	-	-	-	-	-	-	3	-	1	4
MD31XX	PEC2	PROGRAM ELECTIVE II COURSERA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	3
MD32XX	OE1	OPEN ELECTIVE I COURSERA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	4
MM0803A	MDM 3	THEORY OF COMPUTATION	2	1	-	-	30	-	20	10	-	-	-	-	40	-	-	-	2	1	-	3
SE3205	VSEC	DESIGN THINKING -3	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	1	-	1
SE3204	ERI	ENGINEERING RESEARCH AND INNOVATION - I	-	-	2	-	-	-	-	-	-	30	-	70	-	-	-	-	-	-	2	2
Total			**10**	**2**	**8**	**40**	**100**	**10**	**40**	**30**	**0**	**30**	**0**	**70**	**120**	**0**	**60**	**300**	**10**	**2**	**5**	**24**

BOS Chairman **Dean Academics**

Program Elective I:

Subject Code	Subject Name	Subject Code	Subject Name
SE3203A	Fundamental of Machine Learning	**SE3203C**	Information Security and Cryptography
SE3203B	Cloud Computing	**SE3203D**	IoT & Blockchain Technology
ID3001	Basics of Game Development	**ID3002**	Mainframe Techniques

Program Elective II Coursera:

Subject Code	Subject Name	Subject Code	Subject Name
MD3102	Meta Back-End Developer	**MD3119**	Fractal Data Science
MD3106	Microsoft Power BI Data Analyst	**MD3121**	IBM DevOps and Software Engineering
MD3112	Google Cybersecurity	**MD3141**	AWS Cloud Technology Consultant
MD3113	Google Data Analytics	**MD3101**	IBM Full Stack Software Developer
MD3118	IBM Data Science	**MD3147**	Deep Learning.ai Deep Learning
ID3004	DEVOPS AWS SKILL BUILDER-1		

Open Elective I Coursera:

Subject Name	Subject Name
Bucket 1: FinTech (MD3201)	**Bucket 2: Digital Marketing (MD3202)**
FinTech: Finance Industry Transformation and Regulation Specialization	Foundations of Digital Marketing and E-commerce
Digital Transformation in Financial Services Specialization	Adobe Digital Marketing
Financial Markets Specialization	Digital Marketing Operations & Analytics Specialization
FinTech Law and Policy	Social media and Digital Marketing Fundamentals
Decentralized Finance (DeFi): The Future of Finance Specialization	Introduction to Digital Marketing
Bucket 3: Entrepreneurship (MD3203)	Assess for Success: Marketing Analytics and Measurement
Essentials of Entrepreneurship: Thinking & Action	Concepts, Strategies, and Analytics in Performance Marketing and Digital Advertising Specialization

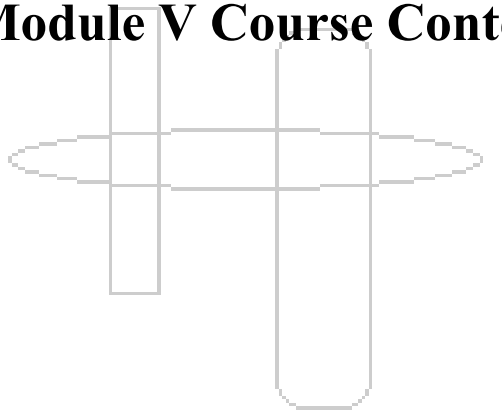
Technology & Entrepreneurship Specialization	Integrated Marketing Communications: Advertising, Public Relations, Digital Marketing and more
Entrepreneurship: Growing Your Business Specialization	Marketing in a Digital World
Skills for a Modern Entrepreneur or Business Owner Specialization	Marketing Analytics Foundation
Innovation & Entrepreneurship - From Design Thinking to Funding	**Bucket 4: Management (MD3204)**
Innovation and entrepreneurship Specialization	Digital Product Management Specialization
	Project Management: Tools, Approaches, Behavioural Skills Specialization
	Foundations of Management Specialization
	Supply Chain Management Specialization

T.Y. B.Tech. Computer Engineering

(Software Engineering)

AY 2026-27

Module V Course Content



FF No.: 654

SE3201: AI IN SOFTWARE ENGINEERING

Teaching Scheme	Examination Scheme
Credits: 4	CP:
Lectures: 3 Hrs/week	PR + CVV: 60 Marks
Practical: 2 Hrs/week	Lab CA:
Tutorial: 0 Hrs/week	ESE (W): 40 Marks

Prerequisites:	
•	• Problem solving and Programming • Design and Analysis of Algorithms • Software Engineering Fundamentals
Course Relevance:	
	Ethical issues in AI-driven software engineering matter because they ensure software systems are fair, safe, and trustworthy. Bias can cause unfair results, transparency makes decisions easier to explain, privacy protects sensitive data, and accountability ensures engineers take responsibility for AI outcomes. Learning these points helps students use AI responsibly in projects, balancing efficiency with fairness and trust.
Course Objectives:	
•	Understand the fundamentals of Artificial Intelligence, Intelligent Agents, Machine Learning, Deep Learning, and intelligent search techniques.
•	Apply AI methods and search algorithms to solve software engineering problems.
•	Use AI tools and techniques across different phases of the Software Development Life Cycle (SDLC).
•	Evaluate AI-based software engineering solutions for improving software quality, productivity, and decision-making.
Course Outcomes:	
	After completion of the course, student will be able to
1.	Explain the fundamentals of AI, Intelligent Agents, and intelligent search techniques.
2.	Apply AI techniques to solve software engineering problems.
3.	Use AI tools across different phases of the Software Development Life Cycle (SDLC).
4.	Evaluate AI-based solutions for improving software quality and productivity.
Section1:	**Topics/Contents**
Introduction to Artificial Intelligence and Intelligent Agents Introduction to Artificial Intelligence (AI), Human Intelligence vs Artificial Intelligence, The Turing Test, Characteristics of Intelligent Agents, Types of Intelligent Agents, AI Environments and Types of Environments, Problem-Solving Approach in AI, Domains and Applications of	

AI, Introduction to Machine Learning (ML), Deep Learning (DL), and Data Science (DS), AI vs ML vs DL vs DS
Case Study: -AI for Everyone / Building AI Projects

Intelligent Search Techniques
Problem Formulation as State Space Search, Problem Characteristics, Search Strategies
Uninformed Search: - Breadth-First Search (BFS), Depth-First Search (DFS), Uniform Cost Search (UCS)
Informed (Heuristic) Search: - Greedy Best-First Search, A* Search
Heuristic Search Techniques: - Generate-and-Test, Hill Climbing, Best-First Search Introduction to Constraint Satisfaction Problems (CSP)

Section2:	**Topics/Contents**

Introduction to AI in Software Engineering
Role of AI in Software Engineering, AI across Software Development Life Cycle (SDLC),
Case Study: - AI Coding Assistants (GitHub Copilot, ChatGPT)
AI in Requirements Engineering
Requirement elicitation using AI, Natural Language Processing (NLP), Requirement classification, Requirement prioritization, Requirement ambiguity detection AI-based documentation generation
Applications: -Chatbots for requirement gathering, Requirement summarization, User story generation
Case Study: -AI-assisted Requirement Analysis
AI in Software Design and Development
AI-assisted software architecture design, Knowledge Representation, AI-assisted UML diagram generation, AI-based code generation, AI code completion, Code optimization, Automated code review, AI pair programming, AI Tools, GitHub Copilot, ChatGPT
Case Study: -Developing an Online Banking System using AI-assisted Development
AI in Software Testing
Automated Test Case Generation, AI-assisted Regression Testing, Bug Detection, Defect Prediction, Software Quality Prediction, Intelligent Debugging AI Tools, SonarQube AI
Case Study: -AI-based Automated Testing
AI in Deployment, Maintenance and Project Management
AI in DevOps (AIOps), Predictive Software Maintenance, AI-based Log Analysis, Intelligent Monitoring, Software Effort Estimation, AI for Project Scheduling and Risk Prediction, Future of Autonomous Software Engineering
Case Study: -AI-powered DevOps Pipeline

Text Books: ***(As per IEEE format)***

1	S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2010. ISBN: 978-0-13-604259-4.
2	E. Rich, K. Knight, and S. B. Nair, *Artificial Intelligence*, 3rd ed. New Delhi, India: Tata McGraw-Hill Education, 2009. ISBN: 978-0-07-008770-5.
3	S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2021

4	M. Virvou and Y. Tanabe, Eds., *Artificial Intelligence-Empowered Software Engineering*. Cham, Switzerland: Springer, 2024.
5	I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2015.
Reference Books: ***(As per IEEE format)***	
1	D. Khemani, *A First Course in Artificial Intelligence*. New Delhi, India: McGraw Hill Education (India), 2013. ISBN: 978-1-25-902998-1.
2	S. Kaushik, *Artificial Intelligence*. New Delhi, India: Cengage Learning India, 2011. ISBN: 978-81-315-1099-5.
3	M. Harman and P. O'Hearn, *Software Engineering and Artificial Intelligence*. Cambridge, UK: Cambridge University Press, 2021.
4	J. Whittle, *Automated Software Engineering: AI Approaches*. Cham, Switzerland: Springer, 2022
URL for Self-Study:	
1	https://nptel.ac.in/courses/106102220
2	https://www.coursera.org/learn/team-software-engineering-with-ai

List of Practical's:

1. Write a Python program to demonstrate intelligent agents by implementing a simple reflex agent (e.g., Vacuum Cleaner Agent).
2. Write Python programs to implement Breadth-First Search (BFS) and Depth-First Search (DFS) for graph traversal.
3. Write a Python program to implement Uniform Cost Search (UCS) for finding the least-cost path.
4. Write a Python program to implement Greedy Best-First Search and A* Search algorithms for shortest path finding.
5. Write a Python program to implement the Hill Climbing algorithm for solving an optimization problem (e.g., 8-Puzzle or 8-Queens).
6. Write a Python program to solve a Constraint Satisfaction Problem (CSP) using backtracking (e.g., Map Coloring or Sudoku).
7. Develop a Python application that uses the OpenAI API/Gemini API to generate Software Requirement Specifications (SRS), user stories, or software documentation.
8. Develop a Python application that generates source code and performs code review using an AI API (ChatGPT/Gemini).
9. Develop a Python application to generate unit test cases and detect bugs in Python programs using an AI API.
10. Mini Project: Develop a Python-based AI-assisted Software Engineering tool (e.g., AI Requirement Analyzer, AI Code Generator, AI Documentation Generator, or AI Test Case Generator) using Streamlit and an AI API.

List of Project areas:

1. AI-Based Software Requirement Analysis and Prioritization System
2. Software Defect Prediction and Bug Classification Using Machine Learning
3. Intelligent Project Effort Estimation and Risk Prediction System
4. Rule-Based Expert System for Software Development Life Cycle (SDLC) Model Selection

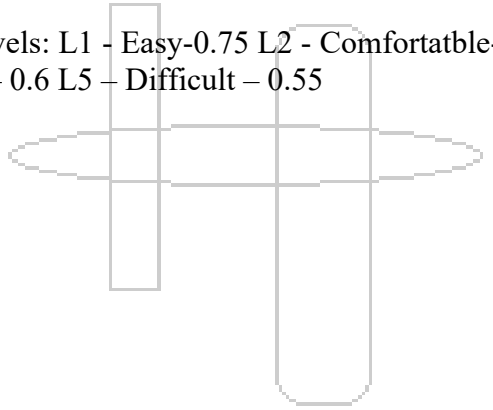
5. AI-Based Automated Test Case Generation and Validation System
6. Software Module Dependency Analysis and Path Optimization Using Graph Search Algorithms
7. Requirement Traceability and Change Impact Analysis System
8. Software Quality Assessment and Reliability Prediction System
9. Intelligent Bug Tracking and Issue Prioritization System
10. AI-Assisted Software Engineering Toolkit *(Requirement Analyzer, UML Generator, Effort Estimator, Defect Predictor, and Test Case Manager)*

CO-PO Mapping

CO	**Program Outcomes (PO)**												**PSO**	
CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1		2					1		2	3	2
CO2	2	3	3	2	3				1	2		2	3	2
CO3	2	3	2	3	3				1	2	1	2	3	2
CO4	2	2	2	2	3	2	1	3	2	2	3	3	2	3
Average	2.25	2.5	2	2.33	2.75	2	1	3	1.33	1.75	2	2.25	2.75	2.25

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55



FF No.: 654

SE3202: AGILE METHODOLOGY

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures: 2 Hrs/week	CVV: 20 Marks
Practical: 2 Hrs/week	Lab CA: 10 Marks
Tutorial: Hrs/week	ESE (W): 40Marks

Prerequisites:	
•	Basics of Software Engineering
Course Relevance:	
	The course relevance of Agile Methodology is driven by its shift from a rigid, "plan-heavy" academic exercise to a dynamic, "industry-standard" practice. Modern degree programs (CS, IT, and Business) integrate Agile to bridge the gap between classroom theory and real-world employment.
Course Objectives:	
•	To introduce characteristics of an agile development process.
•	To understand agile software development process models and plan driven process models.
•	To understand software project characteristics that would be suitable for an agile process.
•	To impart and Identify software project characteristics that would not be suitable for an agile process.
Course Outcomes:	
	After completion of the course, student will be able to
1.	Explain fundamentals of Agile methodology.
2.	Explain agile principles.
3.	Apply Scrum principles.
4.	Apply practices of XP and Incremental design.
Section1:	**Topics/Contents**

Fundamentals of Agile

The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools

Agile Scrum Framework

Introduction to Scrum, Project phases, Agile Estimation, Planning Game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and

retrospective, Daily scrum, Scrum roles Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management.	
Section2:	**Topics/Contents**
Agile Testing The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), Unit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester. **Agile Software Design and Development** Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools.	
Text Books: ***(As per IEEE format)***	
1	K. Schwaber and M. Beedle, *Agile Development with Scrum*, Upper Saddle River, NJ, USA: Prentice Hall.
2	P. Schuh, *Integrating Agile Development in the Real World*, Hingham, MA, USA: Charles River Media.
Reference Books: ***(As per IEEE format)***	
1	A. Cockburn, *Agile Software Development: The Cooperative Game*, 2nd ed. Boston, MA, USA: Addison-Wesley.
2	M. Cohn, *Succeeding with Agile: Software Development Using Scrum*, Boston, MA, USA: Addison-Wesley.
URL for Self-Study:	
1	ScienceDirect - A Decade of Agile Methodologies **:** A comprehensive review of the evolution and effectiveness of Agile from 2001–2010.
2	Coursera - Introduction to Agile Development and Scrum**:** A structured academic course by IBM focusing on practical implementation
3	Study.com - Agile Methodology: Types & Examples**:** Video-based lessons and quizzes for formal self-study

List of Practical's:

1. Study & evaluate the historical and contextual factors that contributed to the rise of Agile software development methodologies.
2. Create & evaluate the definition of user stories and explain their significance within the context of Agile software development methodologies.
3. Create a Scrum Team which defines & identifies the key roles within a Scrum Team based on Real Time Scenario.
4. Apply Design principle and Refactoring to achieve agility based on the above Real Time Scenario.
5. Implement Jira, a project management tool that helps teams plan, track, and manage work for Software Application.
6. Case study (Agile in Software Development vs. Other Industries (Manufacturing, Education, Healthcare)
7. Study and implement automated build tool Docker/ Kubernetes/ OpenShift for an Application.

8. Use Continuous Integration tool such as Jenkins for Software Implementation.
9. Perform Testing activities within an agile project using Jira.

CO-PO Mapping

CO	**Program Outcomes (PO)**												**PSO**	
CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	-	-	-	1	1	2	2	2	1	2	2	1
CO2	2	3	2	2	2	-	-	-	2	1	2	1	3	2
CO3	1	2	3	2	3	1	-	-	3	3	3	2	2	3
CO4	2	2	2	3	3	-	-	1	2	2	2	2	3	2
Average	1.75	2	2.33	2.33	2.66	1	1	1.5	2.25	2	2	1.75	2.5	2

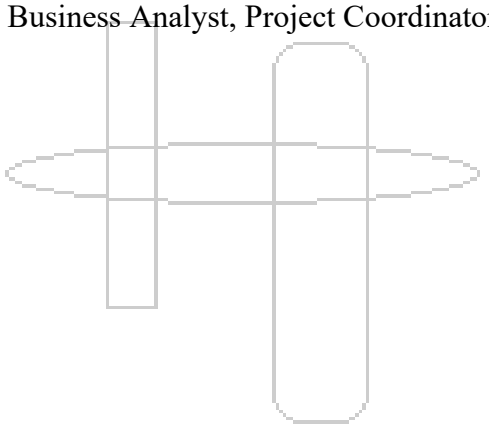
CO Attainment levels:

Future Courses Mapping:

Software Quality Assurance, Cybersecurity, Cloud Computing, DevOps.

Job Mapping:

QA Automation Engineer, Business Analyst, Project Coordinator, Scrum Master, Product Manager.



FF No.: 654

SE3203A: Fundamental of Machine Learning

Teaching Scheme	Examination Scheme
Credits: 4	CP: 40 Marks
Lectures: 3 Hrs/week	CVV: 20 Marks
Practical: 2 Hrs/week	GD/PPT: 20 Marks
Tutorial: Hrs/week	HA: 20 Marks

Prerequisites:	
•	Linear Algebra, Calculus, Probability & Statistics, Fundamentals of Programming and Problem solving.
Course Relevance:	
•	This course provides foundational and advanced knowledge of Machine Learning techniques for analysing data, building predictive models, and solving real-world problems across various domains. It equips learners with practical skills in data preprocessing, model development, evaluation, and implementation using modern Python-based Machine Learning tools and frameworks. The course also emphasizes ethical AI practices, model optimization, and hands-on applications to prepare students for industry, research, and intelligent system development.
Course Objectives:	
•	To understand the fundamentals, types, and applications of Machine Learning.
•	To apply data preprocessing, feature engineering, and model evaluation techniques.
•	To develop and analyze supervised and unsupervised learning models.
•	To implement and optimize Machine Learning solutions using Python tools and real-world datasets.
Course Outcomes:	
	After completion of the course, student will be able to
1.	Explain the fundamentals and applications of Machine Learning.
2.	Perform data preprocessing, feature selection, and exploratory data analysis.
3.	Apply supervised and unsupervised learning algorithms for data analysis and prediction.
4.	Develop and optimize Machine Learning models using Python tools and real-world datasets.
Section 1:	**Topics/Contents**
Introduction to Machine Learning Definition and scope of Machine Learning, types of Machine Learning (supervised, unsupervised, reinforcement learning), real-world applications, machine learning workflow (data collection, preprocessing, training, evaluation), introduction to datasets and features, challenges in Machine Learning. **Data Preprocessing and Exploratory Data Analysis**	

Data cleaning (handling missing values, noise, outliers), data transformation and normalization, feature scaling (Min-Max, standardization), feature selection, basic dimensionality reduction, exploratory data analysis techniques (histograms, box plots, scatter plots), train-test split, cross-validation, bias-variance tradeoff.

Supervised Learning Algorithms

Regression techniques (linear regression, multiple linear regression, polynomial regression), classification techniques (logistic regression, K-Nearest Neighbors, decision trees, Naïve Bayes), model evaluation metrics (accuracy, precision, recall, F1-score, confusion matrix). (Split this into two) classification regression evaluation metrics.

Section 2:	**Topics/Contents**

Unsupervised Learning Algorithms

Clustering techniques (K-Means, hierarchical clustering), dimensionality reduction (Principal Component Analysis), association rule learning (Apriori algorithm), evaluation of clustering models.

Advanced Machine Learning Techniques

Ensemble learning (bagging, boosting, AdaBoost, gradient boosting), Support Vector Machines, introduction to neural networks, basics of deep learning, hyperparameter tuning, overfitting, underfitting, regularization techniques.

Practical Implementation and Applications

Implementation using Python (NumPy, Pandas, Matplotlib, Scikit-learn), model deployment basics, case studies (spam detection, recommendation systems, image classification), ethics in Machine Learning, mini project development using real-world dataset.

Text Books: ***(As per IEEE format)***

1	T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill.
2	C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer
3	E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge, MA, USA: MIT Press, 2020.

Reference Books: ***(As per IEEE format)***

1	A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
2	K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

List of Practical's:

1. Preprocess a real-world dataset by handling missing values, outliers, and applying normalization techniques.
2. Perform exploratory data analysis using statistical and visualization techniques.
3. Implement linear and multiple linear regression models for prediction tasks.
4. Build and evaluate classification models using Logistic Regression and KNN algorithms.
5. Implement Decision Tree and Naïve Bayes classifiers for data classification.
6. Apply K-Means and Hierarchical Clustering techniques for data grouping.
7. Implement Principal Component Analysis (PCA) for dimensionality reduction.
8. Apply ensemble learning techniques such as Bagging, AdaBoost, and Gradient Boosting.
9. Develop prediction models using Support Vector Machines (SVM) and Neural Networks.

10. Mini Project: Develop a real-world machine learning application using Python-based tools and datasets.

List of Project areas:

Project areas in Machine Learning include Predictive Analytics, Healthcare Diagnosis Systems, Smart Agriculture Systems, Fraud Detection Systems, Stock Market Prediction, Recommendation Systems, Sentiment Analysis, Image Classification, Face and Speech Recognition Systems, Natural Language Processing Applications, AI Chatbots, Cybersecurity Analytics, Intrusion Detection Systems, Customer Churn Prediction, Credit Risk Prediction, Smart Traffic and Autonomous Vehicle Systems, Weather and Disease Prediction Models, Energy Forecasting, Smart City Analytics, Fake News Detection, Education Analytics, Human Activity Recognition, Sales Forecasting, IoT Monitoring Systems, Predictive Maintenance, Emotion Recognition, AI Healthcare Monitoring, Recruitment Analytics, Handwritten Recognition, Medical Image Analysis, Spam Detection, Cloud-Based ML Applications, Edge AI Systems, Explainable AI Applications, Smart Energy Systems, Virtual Assistants, and Decision Support Systems.

CO-PO Mapping

CO	**Program Outcomes (PO)**												**PSO**	
CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	-	-	-	-	-	-	1	-	2	-	3	-
CO2	3	3	-	2	-	-	-	-	-	-	2	-	3	-
CO3	3	3	-	2	-	-	-	-	-	-	2	1	3	-
CO4	2	2	3	-	3	-	1	2	2	1	2	-	3	2
Average	2.75	2.5	3	2	3	-	1	2	1.5	1	2	-	3	2

CO Attainment Levels:

Weights for attainment levels:
L1 – Easy – 0.75
L2 – Comfortable – 0.70
L3 – Medium – 0.65
L4 – Somewhat Difficult – 0.60
L5 – Difficult – 0.55

Future Courses Mapping:

Deep Learning, Artificial Intelligence, Data Science, Computer Vision, Natural Language Processing, Reinforcement Learning, Big Data Analytics, Cloud-Based AI Systems, Explainable AI, Edge AI, Robotics, Intelligent Systems, MLOps, and similar courses.

Job Mapping:

Machine Learning Engineer, Data Scientist, AI Engineer, Deep Learning Engineer, Data Analyst, Business Intelligence Analyst, NLP Engineer, Computer Vision Engineer, AI Researcher, Software Engineer, Predictive Analytics Specialist, MLOps Engineer, Research Scientist, and Application Developer.

FF No.: 654

SE3203B: CLOUD COMPUTING

Teaching Scheme	Examination Scheme
Credits: 4	CP: 40 Marks
Lectures: 3 Hrs/week	CVV: 20 Marks
Practical: 2 Hrs/week	GD/PPT: 20 Marks
Tutorial: 0 Hrs/week	HA: 20Marks

Prerequisites:	
•	Fundamental of Computer Network
Course Relevance:	
	Cloud computing is a **foundational pillar** of modern technology, making its study highly relevant for anyone looking to enter the IT sector.
Course Objectives:	
•	Explore Storage and Services of cloud infrastructure.
•	Understand layers and cloud platform for various services.
•	Manage and secure the cloud for risk.
•	Learn to design applications using **microservices** and **API-first** patterns rather than single, monolithic blocks.
Course Outcomes:	
1.	Explain the fundamentals of cloud computing infrastructure, architecture, and service models.
2.	Analyze and apply cloud-based solutions for business and scientific applications using various cloud platforms
3.	Create SOA & cloud services.
4.	Analyze inter-cloud management, storage & security
Section1:	**Topics/Contents**

Introduction to Cloud Computing & Architecture:
History and Evolution of Cloud Computing, Need and Characteristics of Cloud Computing, Cloud Computing Architecture, Advantages and Disadvantages, Cloud Service Providers Overview, Cloud Deployment Models – Public, Private, Hybrid, Introduction to Cloud Security, Scalability, Time to Market, Cloud Architecture Layers: Infrastructure, Platform, and Software as a Service (IaaS, PaaS, SaaS),Virtualization and its Benefits, Virtual Private Cloud (VPC),Service Level Agreements (SLA) and Migration to the Cloud.
Case Studies: Features of Cloud Platforms (e.g., AWS, GCP)
Physical & Data Link Layer:
Evaluating Cloud Platforms, Overview of Public Cloud Providers: Amazon Web Services (AWS),Microsoft Azure, Google Cloud Platform (GCP),Salesforce, Private Cloud Technologies:

Azure Stack, Storage and Security Services ,Managing and Securing Cloud Services, Cloud Cost Models and Resource Management, Service-Oriented Architecture (SOA) and Cloud. **Case Studies:** Billing management tool.	
Section2:	**Topics/Contents**
Security and Risk in Cloud Computing (Azure): Introduction to Cloud Security and Risk, Key Principles of Information Security, Common Security Threats: Insider Threats, Data Loss and Exposure, Organizational and Criminal Threats, Mitigation Strategies and Risk Management, Security Standards and Digital Forensics and Incident Response, Application Interfaces and Shared Technology Risks, Disaster Recovery. **Case Studies:** AWS monitoring, Troubleshooting in the cloud. **Cloud Applications and Deployment:** Cloud-Based Application Types: Desktop, Distributed, Web-Based, and Cloud Native, Developing Cloud-Ready Applications, Patterns and Tools, Docker, Docker Hub, Application Migration Challenges and Solutions, Technical and Risk Preparation for Migration, DevOps and Site Reliability Engineering (SRE) in Cloud, Deployment on Commercial Cloud: Google App Engine, Microsoft Azure, AWS EC2. **Case Studies:** Deploy a sample application on cloud (using Docker or cloud provider)	
Text Books: ***(As per IEEE format)***	
1	Mastering Cloud Computing Foundations and Applications Programming" by Rajkumar Buyya
2	Enterprise Cloud Computing – Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010
Reference Books: ***(As per IEEE format)***	
1	Cloud Security by Ronald Krutz and Russell Dean Vines, Wiley – India, 2010
URL for Self-Study:	
1	https://onlinecourses.nptel.ac.in/noc22_cs20/preview

List of Practical's:

1. To study basic Linux commands.
2. Case study on Amazon EC2/ Microsoft Azure.
3. Study and installation of Storage as Services on cloud (S3 service)
4. Implementation of identity access management. (IAM Service)
5. Implementation of Virtual Private cloud(VPC Service)
6. To create and access VM instances and demonstrate various instance.
7. To create and access VM instances and demonstrate various components such as EC2, S3, Simple DB, DynamoDB. Technology: AWS
8. Deploy web applications on commercial cloud. Technology: Google appEngine/ Windows Azure 2
9. Study and explore – case study with devops and site reliability engineering
10. Disaster Recovery in the cloud.
11. To create your EC2 instance using Python and Boto3 Library.
12. Write a program in Python for S3 bucket in Boto3 Library.

13. Dockerized Application Deployment using Jenkins
14. Course Project

List of Project areas:
Serverless Web Applications: Build apps using AWS Lambda, Azure Functions, or Google Cloud Functions that scale automatically without manual server management.
Infrastructure-as-Code (IaC): Use tools like Terraform or AWS CloudFormation to provision entire environments (VPCs, servers, databases) via scripts.
CI/CD Pipelines: Design automated software release pipelines using GitHub Actions, AWS CodePipeline, or Azure DevOps to manage code from testing to production.
Cloud Migration: Practice migrating an on-premises database (like MySQL) or an existing virtual machine to a cloud platform like AWS

CO-PO Mapping

CO	Program Outcomes (PO)												PSO	
CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	–	–	2	–	–	1	1	–	1	2	3	-
CO2	2	2	3	2	2	–	1	1	3	2	1	2	3	-
CO3	3	2	–	–	–	–	–	1	–	1	–	1	3	-
CO4	3	3	–	2	–	–	–	–	–	–	1	–	3	2
Average	3	2.25	3	2	2	--	1	1	2	1.5	1	1.75	3	2

CO Attainment levels:
Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

Future Courses Mapping:

- Microservices Architecture: Transitioning from single apps to distributed systems.
- DevOps & Site Reliability Engineering (SRE): Mastering CI/CD pipelines, Kubernetes, and "self-healing" infrastructure.
- Infrastructure as Code (IaC): Learning to write code (Terraform, Ansible) to manage global data centers.
- Multi-Cloud Strategy: Learning how to bridge AWS, Azure, and Google Cloud for corporate redundancy.

Job Mapping:

- Cloud Solutions Architect: Designs the overall structure (network, storage, compute) to meet business needs.
- Cloud Engineer: Implements the architect's plan; handles the technical setup and maintenance of cloud environments.
- Network Architect: Focuses specifically on the "pipes" connecting different cloud services and on-premises data centers.

FF No.: 654

SE3203C: Information Security and Cryptography

Teaching Scheme	Examination Scheme
Credits: 4	CP: 40
Lectures: 3 Hrs/week	CVV: 20
Practical: 2 Hrs/week	HA: 20
Tutorial: 0 Hrs/week	PPT:20

Prerequisites:	
	Basic understanding of computer Networks
Course Relevance:	
	This course introduces essential concepts of information security and cryptography used to protect data in modern digital systems. It helps students understand secure communication, encryption techniques, and threat mitigation, while building a strong foundation for advanced cybersecurity studies and careers.
Course Objectives:	
●	**To understand** the fundamental concepts of information security, including security goals, threat models, and security standards.
●	**To analyze** classical and modern encryption techniques, including symmetric and asymmetric cryptographic methods.
●	**To design and implement** secure encryption systems using block ciphers such as DES and AES along with appropriate modes of operation.
●	**To apply** public key cryptography, key management techniques, and cryptographic protocols to ensure secure communication, authentication, and data integrity.
Course Outcomes:	
	After completion of the course, student will be able to
1.	learn fundamental security concepts, goals, and classical cryptographic techniques.
2.	Apply symmetric key cryptographic algorithms such as DES and AES.
3.	Implement asymmetric cryptographic techniques and key exchange mechanisms.
4.	Analyze authentication mechanisms and security protocols such as SSL/TLS and IPSec.
5.	Evaluate cryptographic applications and identify implementation vulnerabilities.
6.	Analyse cryptographic attacks and propose suitable security mechanisms.
Section1:	**Topics/Contents**
Security Fundamentals and Cryptographic Principles Introduction to information security, Security Goals: Confidentiality, Integrity, Availability, Security policies. Threat models, and security services. Security Standards **Encryption Algorithms** Introduction, Classical techniques: substitution and transposition ciphers, an overview of symmetric and asymmetric cryptography. Key space, brute-force attacks, and steganography	

Encryption Methods: Symmetric, Asymmetric, Cryptography, Substitution Ciphers. Transposition Ciphers, Block Ciphers and methods of operations, **Symmetric Key Cryptography and Block Ciphers** Symmetric encryption techniques -block cipher structures and modes of operation. Detailed coverage of Data Encryption Standard (DES), Triple DES, and Advanced Encryption Standard (AES) -working principles, design criteria, and performance aspects. Security issues such as weak keys and cryptanalysis techniques.
Section2: Topics/Contents
Asymmetric Cryptography and Key Management Public key cryptography concepts, RSA algorithm- key generation, encryption, and decryption processes. Diffie-Hellman key exchange is discussed for secure key distribution. Cryptographic hash functions such as SHA and MD5, Authentication Codes (MAC), data integrity and authentication **Authentication and Cryptographic Protocols** Authentication mechanisms and protocols: Kerberos and X.509, Key Infrastructure (PKI) and digital certificates. Secure communication protocols: SSL/TLS, IPSec, wireless protocol WPA2/A3, architecture, and secure email systems, PGP and S/MIME **Cryptographic Applications and Case Studies** Applications of cryptography in real-world systems: digital signatures, secure multiparty computation, single sign-on systems, and secure financial transactions. Implementation-level vulnerabilities such as cross-site scripting. **Cryptographic Attacks and Security Mechanisms** Cryptanalysis techniques: Brute-force, replay, and side-channel attacks. Fundamental concepts of intrusion detection and basic security mechanisms relevant to protecting cryptographic systems.

Text Books: ***(As per IEEE format)***	
1	William Stallings: "Network Security Essentials: Applications and Standards"
2	Mark Ciampa "Security+ Guide to Network Security Fundamentals"
Reference Books: ***(As per IEEE format)***	
1	Charlie Kaufman, Radia Perlman, Mike Speciner— "Network Security: Private Communication in a Public World
2	Brian Carrier. "File System Forensic Analysis
URL for Self-Study:	
1	NPTEL: Cryptography and Network Security (Course Code: 106105031) https://nptel.ac.in/courses/106105031
2	NPTEL: Introduction to Cryptology (Course Code: 106105042) https://onlinecourses-archive.nptel.ac.in
3	SWAYAM/NPTEL: Cryptography and Network Security (Code: noc22_cs90) https://onlinecourses.nptel.ac.in

List of Practical's: (Python / C / Java**)**

1. Implementation of basic encryption and decryption techniques
2. Study Authentication and authorization and Demonstration of token generation
3. Implementation of AES-based symmetric encryption
4. Implementation of DES-based symmetric encryption
5. Implementation of RSA algorithm

6. Implementation of hash functions (MD5/SHA) for password security
7. Implementation of Diffie-Hellman key exchange
8. Implementation of basic attacks and mitigation techniques
9. Course project based on cryptographic applications/Security

List of Project areas:

- Cryptanalysis and attack simulation (brute-force, side-channel, replay attacks)
- Secure email systems using PGP or S/MIME
- Web application security focusing on vulnerabilities like cross-site scripting (XSS)
- Secure cloud storage systems using encryption techniques
- Lightweight cryptography for IoT device

CO-PO Mapping

CO	**Program Outcomes (PO)**												**PSO**	
CO \ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO12	PSO1	PSO2
CO1	3	2	-	-	-	1	-	1	1	-	2	1	3	-
CO2	3	3	1	2	1	-	-	-	-	-	2	1	3	-
CO3	3	3	2	2	2	-	-	-	-	-	2	1	3	-
CO4	2	2	3	1	3	1	1	2	2	1	2	1	3	2
Average	**2.75**	**2.5**	**2**	**1.67**	**2**	**1**	**1**	**1.5**	**1.5**	**1**	**2**	**1**	**3**	**2**

CO Attainment levels:
Weights for attainment levels: L1–Easy (0.75), L2–Comfortable (0.70), L3–Medium (0.65), L4–Somewhat Difficult (0.60), L5–Difficult (0.55).

Future Courses Mapping:
Network Security, Cyber Security, Cloud Computing, Distributed Systems, Wireless and Mobile Networks.

Job Mapping:
Cybersecurity Analyst, Network Security Engineer, Software Developer, Application Developer, IT Engineer.

FF No.: 654

MM0803A: Theory of Computation

Teaching Scheme	Examination Scheme
Credits: 3	CP: 30 Marks
Lectures: 2 Hrs/week	GD/PPT: 20 Marks
Practical: Hrs/week	HA: 10 Marks
Tutorial: 01 Hrs/week	ESE (W): 40Marks

Prerequisites:	
●	Basic understanding of Graph Theory, Programming Logic, Set theory
Course Relevance:	
	Compiler Design and Parsing: Concepts like Context-Free Grammars (CFG) are essential for parsing and constructing compilers for programming languages. **Pattern Recognition (Regex):** Finite Automata (DFA/NFA) are used to design regular expressions for lexical analysis, pattern matching, and text processing tools. **Algorithm Optimization:** By understanding formal computational models (Turing machines), students learn to design more efficient and scalable algorithms. **Software Verification:** Automata theory assists in verifying the correctness of system behavior, which is important for critical software systems. **Core Foundation:** It provides the theoretical basis for computer science, essential for advanced studies in AI, formal verification, and cryptography.
Course Objectives:	
1.	Understand and design finite automata, regular expressions, and context-free grammars for language recognition.
2.	Analyze the properties of different language classes (Regular, Context-Free, Recursive, Enumerable).
3.	Construct Pushdown Automata and Turing Machines to solve computational problems.
4.	Classify problems into complexity classes like P, NP, and NP-complete.
Course Outcomes:	
	After completion of the course, student will be able to
1.	Construct Finite Automata (DFA/NFA) for regular languages and perform conversions between them.
2.	Build Pushdown Automata for Context-Free Languages.
3.	Design Turing Machines for specific, decidable problems.
4.	Prove properties of languages using Pumping Lemmas.
Section1:	**Topics/Contents**

Strings, Alphabet, Language, Operations, Finite State Machine, definitions, Finite automation model, Acceptance of a String and Languages, Non-Deterministic Finite Automation, Deterministic Finite Automation, Equivalence between NFA & DFA, Conversion from NFA into DFA, minimization of FSM, Equivalence between two FSM's, Moore & Melay machines. Regular Sets, Regular Expressions, Identity Rules, Manipulation of Regular Expressions, Equivalence between RE & FA, Interco version, Pumping Lemma, Closure properties of Regular Sets (not proofs), Regular Grammars, Right Linear Grammars, Left Linear Grammars, Equivalence between Regular Linear Grammar & FA, Interco version between RE & RG. Context Free Grammar, Derivation Tree, Chomsky Normal Form, Greibach Normal Form, Push Down Automata, Definition, Model of Acceptance of CFL, Equivalence of CFL & PDA, Interco version, Enumeration of properties of CFL (no proofs).

Section2:	**Topics/Contents**
	Turing Machine, Definition, Model of acceptance, Design of TM Computable functions. Recursive enumerable language, Church's hypothesis, Counter Machines, Types of TM. TM in Quantum Computing Chomsky Hierarchy of languages, Linear Bounded Automata and Context Sensitive Language, Introduction to DCFL & DPDA, LR (0) grammar, Decidability of problems. Undecidability, Properties of recursive & non-recursive enumerable languages, Universal Turing Machine, *Post Machines* as an alternative model of computation Post-correspondence problem, Church–Turing Thesis. Introduction to recursive function theory.
Text Books: ***(As per IEEE format)***	
1	J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2006.
2	P. Linz, *An Introduction to Formal Languages and Automata*, 7th ed. Burlington, MA, USA: Jones & Bartlett Learning, 2023.
Reference Books: ***(As per IEEE format)***	
1	J. C. Martin, *Introduction to Languages and the Theory of Computation*, 4th ed. New Delhi, India: McGraw-Hill Education (India), 2011.
2	H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, 2nd ed. New Delhi, India: Prentice Hall of India, 1998.
3	M. Davis, *Computability and Unsolvability*. New York, NY, USA: Dover Publications, 1982.
URL for Self-Study:	
1	https://onlinecourses.nptel.ac.in/noc22_cs63/preview
2	https://ocw.mit.edu/courses/18-404j-theory-of-computation-fall-2020/
3	https://www.coursera.org/courses?query=theory%20of%20computation

List of Tutorial's:

1. Simulating a DFA to recognize strings starting with 'a' and ending with 'b'.
2. Designing a PDA that accepts well-formed parentheses.
3. Creating a Turing Machine that determines 2's complement of a binary string.
4. Using Lex to implement a tokenizing program.
5. LR Parsing and Decidability
6. 6.Simulating NFA with epsilon-transitions.
7. Undecidability and Universal Computation
8. Advanced Models of Computation

9. Constructions of Post Machines.
10. Use of AI tools: FLAP and the Automata Simulator

Sample List of course projects:

1. Design and Simulation of a Deterministic and Non-Deterministic Finite Automata (DFA & NFA)
2. NFA to DFA Conversion and DFA Minimization Tool
3. Regular Expression to Finite Automata Converter
4. Pushdown Automata (PDA) Simulator for Context-Free Languages
5. Context-Free Grammar (CFG) Parser and String Validator
6. Turing Machine Simulator for Basic Computational Problems
7. Lexical Analyzer Using Finite Automata
8. Implementation of Moore and Mealy Machine Converters
9. Pumping Lemma Verification Tool for Regular Languages
10. Theory of Computation Learning and Visualization Platform
11. Interactive Chomsky Hierarchy Classifier for Formal Grammars
12. CYK Parsing Algorithm for Context-Free Grammar Membership Testing
13. Universal Turing Machine Simulator
14. Finite Automata-Based Pattern Matching System
15. Visualization Tool for Computability and Decidability Problems

CO-PO Mapping

CO	**Program Outcomes (PO)**												**PSO**	
CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	-	-	2	3	-	-	-	-	1	3	-
CO2	3	3	-	2	-	2	-	-	-	-	-	1	3	-
CO3	3	3	-	2	-	-	-	-	-	-	2	1	3	-
CO4	3	2	3	-	3	-	-	-	-	-	-	2	3	2
Average	3	2.75	2.5	2	3	2	3	-	-	-	2	1.25	3	2

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

Future Courses Mapping:

Natural Language Processing (NLP): Relies on Context-Free Grammars and pushdown automata for syntactic analysis.

Compiler Optimization & Software Engineering: Utilizes automata theory for regular expressions and state minimization.

Cryptography & Security: Leverages computational complexity and decidability to determine the security of encryption algorithms.

Quantum Computing: Explores advanced models beyond traditional Turing machines.

Job Mapping:

Backend Software Engineer**:** Utilizing state machines to manage complex UI workflows or system states (e.g., using libraries like XState),

Algorithm Designer/Developer**,** Compiler Designer**,** Formal Verification Engineer**,** Cryptography/Cybersecurity Specialist**,** Academic /Researcher**.**

FF No.: 654

SE3205: Design Thinking 3 and 4

Teaching Scheme	**Examination Scheme**
Credits : 1	CP: Marks
Lectures : Hrs/week	GD/PPT/HA: Marks
Practical : Hrs/week	MSE (W/O):30 Marks
Tutorial : 1 Hrs/week	ESE (W/O):70 Marks

Prerequisites:	
●	Course Prerequisites: Problem Based Learning, Project Centric Learning Course Relevance:
Course Objectives:	
●	To provide ecosystem for students and faculty for paper publication and patent filing
Course Outcomes:	**[Publication of paper or patent]**
	After completion of the course, student will be able to
1.	Understand the importance of doing Research
2.	Interpret and distinguish different fundamental terms related to Research
3.	Apply the methodology of doing research and mode of its publication
4.	Write a Research Paper based on project work
5.	Understand Intellectual property rights
6.	Use the concepts of Ethics in Research, Understand Entrepreneurship and Business Planning

Section1:	**Topics/Contents**
What is research? Importance of Paper Publication and Patents Importance of Paper Publication and Patents Structure of Paper Journal Publication Publication in conference Literature Review Research Paper Writing Journal Ratings and Evaluation How to rate a Journal? Intellectual property (IP) Research Ethics Entrepreneurship	
Section2:	**Topics/Contents**
Structure of the paper Journal List (Top 50 Journals) Selection of the journal Use of various online journal selection tools Plagiarism checking Improving contents of the paper Patent drafting Patent search Filing of patent Writing answers to reviewer questions Modification in manuscript Checking of publication draft	

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)												PSO	
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	1	1	1	1	-	-	-	-	-	-	1	1	2
CO2	1	1	1	1	1	-	-	-	-	-	-	1	1	2
CO3	2	2	3	3	2	2	1	2	2	3	-	1	1	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	1	3
CO5	1	1	1	1	1	-	-	-	-	-	-	1	1	2
CO6	2	2	2	2	2	2	1	3	2	3	-	1	1	3
Avg	1.7	1.7	1.8	1.8	1.7	2.0	1.0	2.3	2.0	3.0	1.0	1.0	1.0	2.5

FF No.: 654

SE3204: Engineering Research & Innovation

Teaching Scheme	Examination Scheme
Credits : 2	CP: Marks
Lectures : Hrs/week	GD/PPT/HA: Marks
Practical : 2 Hrs/week	MSE (W/O):30 Marks
Tutorial : Hrs/week	ESE (W/O):70 Marks

Prerequisites:	
●	Course Prerequisites: Research Based Learning Course Relevance: Research Centric Learning (RCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based research projects, curriculum can be enriched with semester long Engineering Research and Innovation courses, in which students can investigate socially relevant problems using various technologies and research methods from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such research problems, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the research domain, thereby achieving research centric learning. Modern world sustained and advanced through the successful completion of research and innovation. In short, if students are prepared for success in life, we need to prepare them for a research-based world. It is a style of active learning and inquiry-based learning. Research centric learning will also redefine the role of teacher as mentor in the learning process. The RCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.
Course Objectives:	
●	To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
●	To Evaluate alternative approaches, and justify the use of selected tools and methods,
●	To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
●	To engage students in rich and authentic learning experiences.
●	To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
●	To develop an ecosystem to promote entrepreneurship and research culture among the students
Course Outcomes:	

	After completion of the course, student will be able to
1.	Identify a real-life research problem or gap from societal and technological need point of view
2.	Choose and compare alternative research methodologies to select the most feasible one
3.	Analyze and synthesize the identified research problem from a technological and scientific perspective.
4.	Design a reliable and scalable research methodology/experimental approach to meet the challenges.
5.	Evaluate the research outcomes based on the criteria specified
6.	Inculcate a long life learning and research attitude towards the societal problems

Section1:	**Topics/Contents**

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ERI, laboratory course contents of "Engineering Research" are designed as a ladder to extend connectivity of research methodologies and emerging technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Research Databases and Literature Repositories (IEEE Xplore, Scopus, Google Scholar), Patent Search and IPR Tools, Statistical and Data Analysis Tools, Computational Biology (Biomedical and Bioinformatics), Robotics and Automation, Industry 4.0 (Artificial Intelligence, Machine Learning, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

There should be a team/group of 4-5 students. A supervisor/mentor teacher assigned to individual groups. It is useful to group students of different abilities and nationalities together.

Selection of Research Problem/Topic:

● Students must focus on initiating the task/idea. The idea of inception and consideration shall be from following areas as a real world problem:

-Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.

● This is the sample list to start with. Faculty and students are free to include other ar
eas which meet society's requirements at large.

● The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.

● A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.

● By exemplarity, a problem needs to refer to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in RCL:

- Teacher is not the source of solutions rather he will act as the facilitator and mentor.
- To utilize the principles of problem solving, critical thinking and metacognitive skills of the students.
- To make the group aware about time management.
- Commitment to devote the time to solving student's technical problems and interested in helping students to empower them better.

Student's Role in RCL:

- Students must have ability to initiate the task/idea, they should not be mere imitators.
- They must learn to think.
- Students working in RCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, instead of passively receiving instruction.
- Students in RCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in RCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What do they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution.

  How effective is …….?

  How strong is the evidence for ……?

  How clear is ……?

  - What are the justifications for thinking? Why is the method chosen?
  - What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is an integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- · How to prepare for the search? How to carry out research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on the following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

Section2: Topics/Contents

ERI Sample Research Case Studies: -

With the adaptation of industry communication standards and modern research tools and platforms, the following research problems can be taken up:

1) Research on soil moisture prediction and detection using sensor data and machine learning
2) Research on temperature monitoring and prediction using IoT sensors and analytics
3) Research on pressure sensing and anomaly detection techniques for industrial applications
4) Research on early smoke and fire detection using sensor fusion and AI
5) Research on human motion detection and activity recognition using sensors
6) Research on collision detection and avoidance systems using sensor fusion
7) Research on acoustic and sound-based anomaly detection using machine learning

...**not limited to.....Faculty and students are free to include other areas which meet society's requirements at large.**

Text Books: ***(As per IEEE format)***

1	*Research Methodology: Methods and Techniques. By C.R. Kothari. New Age International Publishers. ISBN:978-8122415223; 2004*
2	*Research Methodology: A Step-by-Step Guide for Beginners. By Ranjit Kumar, SAGE Publications. 2019.*
3	*Practical Research: Planning and Design. By Paul D. Leedy, Jeanne Ellis Ormrod, Pearson. 2019*
4	*The Craft of Research. By Wayne C. Booth, Gregory G. Colomb, Joseph M. Williams. University of Chicago Press. 2016*

Reference Books: ***(As per IEEE format)***

1	Creswell J.W.: Research Design: Qualitative, Quantitative and Mixed Methods Approaches. Sage Publications. 2014.
2	Research project management core textbook, second edition, Indian Edition, by Gopalan.
3	The Craft of Research. By Wayne C. Booth, Gregory G. Colomb & Joseph M. Williams

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)												PSO	
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	2					3		2	2	3	3
CO2	2	2	3	2	2		2		3		2	2	3	3
CO3	2	2	3	2	3		2		3	1	2	2	3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3
CO5	2	2	3	2	3	2			3		2	2	3	3
CO6	2	2	3	3	2				3		3	2	3	3
Avg	2.0	2.0	2.83	2.83	2.6	2.5	2.0	2.0	3.0	1.0	2.16	2.0	3.0	3.0