St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

St. JOHNS COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

Yerrakota, YEMMIGANUR - 518360, Kurnool Dt,, Andhra Pradesh, India

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ENGINEERING

Syllabus

(SJCET R25 Regulations)

for

M.Tech (Regular-Full time)

(Effective for the students admitted into I year from the Academic Year 2025-26 onwards)

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

M.TECH-COURSE STRUCTURE– R25

(Applicable from the academic year 2025-26 onwards)

I M.Tech – I Semester

SNO	Course Code	Course Title	L/D	T	P	Credits
1	25G3D58101	Advanced Data Structures & Algorithms	3	0	0	3
2	25G3D25102	Software Architecture Design Patterns	3	0	0	3
3	25G3D25103a 25G3D25103b 25G3D25103c	Program Elective – I Software Project Control & Monitoring Software Metrics and Quality Engineering Advanced Software Engineering	3	0	0	3
4	25G3D25104a 25G3D25104b 25G3D25104c	Program Elective – II AI-Driven Software Engineering Cloud Computing Secure Software Development	3	0	0	3
5	25G3D58105	Advanced Data Structures & Algorithm Lab	0	0	4	2
6	25G3D25106	Software Architecture and Design Patterns Laboratory	0	0	4	2
7	25G3DRM101	Research Methodology and IPR	2	0	0	2
8	25G3D25107	Software Process Automation Tools	0	1	2	2
9	25G3DAC101a 25G3DAC101b 25G3DAC101c	Audit Course – I English for Research Paper Writing Disaster Management Essence of Indian Traditional Knowledge	2	0	0	2
						20

I M.Tech – II Semester

SNO	Course Code	Course Title	L/D	T	P	Credits
1	25G3D25201	Software Testing and Verification	3	0	0	3
2	25G3D25202	Advanced Algorithms and Optimization Techniques	3	0	0	3
3	25G3D25203a 25G3D25203b 25G3D25203c	Program Elective – III 1. Intelligent Software Systems (new) 2. Enterprise Application Development 3. Model-Driven Software Engineering	3	0	0	3
4	25G3D25204a 25G3D25204b 25G3D25204c	Program Elective – IV 1. Machine Learning in Software Engineering Processes 2. Distributed Systems 3. DevOps and Continuous Engineering	3	0	0	3
5	25G3D25201P	Software Testing and Verification Laboratory	0	0	4	2
6	25G3D25202P	Advanced Algorithms and Optimization Techniques Laboratory	0	0	4	2

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

7	25G3D58205	Quantum Technologies And Applications	2	0	0	2
8	25G3D25206	Comprehensive Viva Voce	0	1	2	2
9	25G3DAC201a 25G3DAC201a 25G3DAC201a	Audit Course – II Pedagogy Studies Personality Development Through Life Enlightenment Skills Yoga For Stress Management	2	0	0	0
Total						20

****Students have to undergo an Industry Internship after I Year II Semester for duration of 6 to 8 weeks**

M.Tech-SEMSTER - III

S.No.	Course code	Course Name	Category	Hours per			Credits
				L	T	P	
1.	25G3D25301a 25G3D25301b 25G3D25301c	**Program Elective – V** 1. Service Oriented Architecture 2. Cyber-Physical Systems 3. AI-Enabled Software Quality Assurance	PE	3	0	0	3
2.		**Open Elective-I**	OE	3	0	0	3
3.	25G3D25303	Dissertation Phase – I	PR	0	0	20	10
4.	25G3D25304	Industry Internship		0	0	0	2
5.	25G3D25305	Co- Curricular Activities		0	0	0	1
Total							**19**

Open Elective-I

S.No.	Course Code	Course Name	Offered by the Dept.
1	25G3D20302	Green Buildings	Civil
2	25G3D93203b	Road Safety Engineering	
3	25G3D06302	IoT and its Applications	ECE
4	25G3D07302	Photovoltaic Systems	EEE
5	25G3D15302	Integrated Product Design and Development	ME
6	25G3DOE301a	Advanced Numerical Methods and Computational Mathematics	Mathematics
7	25G3DOE301b	Mathematics for Machine Learning and Data Science	
8	25G3DOE301c	Statistical Learning Theory and Mathematical Foundations of AI	
9	25G3DOE301d	Chemistry Of Nanomaterials And Applications In Engineering	Chemistry
10	25G3DOE301e	Photonics For Engineers	Physics

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

M.Tech IV-SEMESTER

S.No.	Corse Codes	Course Name	Category	Hours per			Credits
				L	T	P	
1.	25G3D25401	Dissertation Phase – II	PR	0	0	32	16
Total							**16**

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ADVANCED DATA STRUCTURES & ALGORITHMS

I MTech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
25G3D58101	Engineering Science	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100

Course Objectives:

To understand concepts of dictionaries and hash tables,

- To implement lists and trees,
- To analyze usage of B trees, Splay trees and 2-3 trees,
- To understand the importance of text processing and computational Geometry,

Course Outcomes: A student after completion of the course will be able to

CO1: Illustrate the working of the advanced tree data structures and their applications
CO2: Understand the Graph data structure, traversals and apply them in various contexts,
CO3: Use various data structures in the design of algorithms
CO4: Analyse algorithms with respect to space and time complexities
CO5: Design new algorithms

UNIT-1
Dictionaries : Definition, Dictionary Abstract Data Type, Implementation of Dictionaries, Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing,
UNIT-1I
Skip Lists : Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists, Trees: Binary Search Trees (BST), AVL Trees, Red Black Trees: Height of a Red Black Tree, Red Black Trees Bottom-Up Insertion, Top-Down Red Black Trees, Top-Down Deletion in Red Black Trees, Analysis of Operations
UNIT-III
2-3 Trees , Advantage of 2-3 trees over Binary Search Trees, Search and Update Operations on 2-3 Trees, Analysis of Operations, B-Trees: Advantage of B- trees over BSTs, Height of B-Tree, Search and Update Operations on 2-3 Trees, Analysis of Operations, Splay Trees: Splaying, Search and Update Operations on Splay Trees, Amortized Analysis of Splaying,
UNIT-IV
Text Processing: Sting Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, TheHuffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem
UNIT-V
Computational Geometry: One Dimensional Range Searching, Two Dimensional Range Searching, Constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quadtrees, k-D Trees

Text books:

1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh 2nd Edition Universities Press
2. Computer Algorithms/C++ Ellis Horowitz, SartajSahni, SanguthevarRajasekaran2nd Edition University Press

Reference Books:

1. Data Structures and program design in C, Robert Kruse, Pearson Education Asia
2. An introduction to Data Structures with applications, Trembley & Sorenson, McGraw Hill 3, The Art of
3. Computer Programming, Vol,1: Fundamental Algorithms, Donald E Knuth, Addison-Wesley, 1997,
4. Data Structures using C & C++: Langsam, Augenstein&Tanenbaum, Pearson, 1995
5. Algorithms + Data Structures &Programs:,N,Wirth, PHI
6. Fundamentals of Data Structures in C++: Horowitz Sahni& Mehta, Galgottia Pub,
7. Data structures in Java:, Thomas Standish, Pearson Education Asia

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Online Links:

1. https://www,tutorialspoint,com/advanced_data_structures/index,asp
2. http://peterindia,net/Algorithms,html 3, Abdul Bari,1, Introduction to Algorithms (youtube,com)

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ARCHITECTURE DESIGN PATTERNS

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25102	Professional Core	3	0	0	3	40	60	100

Course Objectives:

The main objective is to introduce the student to architecture of software and design Patterns, Upon completion of this course the student will get an idea on envisioning architecture, creating architecture, analyzing architecture,

- Understand the creational and structural patterns
- Be capable of applying his knowledge to create an architecture for given application,
- Be able to explain the role of analysing architectures,
- Be able to identify different structural patterns,

Course Outcomes: At the end of the course, Student will be able to

CO1:To understand the concept of patterns and the Catalog,

CO2: To discuss the Presentation tier design patterns and their affect on: sessions, client access, validation and consistency,

CO3: To understand the variety of implemented bad practices related to the Business and Integra-tion tiers,

CO4: To highlight the evolution of patterns,

CO5: To learn how to add functionality to designs while minimizing complexity

,

UNIT-I
Envisioning Architecture: The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views, Creating an Architecture: Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture,
UNIT-II
Analyzing Architectures: Architecture Evaluation, Architecture design decision making, ATAM, CBAM, Moving from one system to many: Software Product Lines, Building systems from off the shelf components, Software architecture in future,
UNIT-III
Patterns: Pattern Description, Organizing catalogs, role in solving design problems, Selection and usage, Creational and Structural patterns: Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight,
UNIT-1V
Behavioural patterns: Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor,
UNIT-V
Case Studies: A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development

Text books:

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick Kazman, Pearson Education, 2003,
2. Design Patterns, Erich Gamma, Pearson Education,

Reference Books:

1. Beyond Software architecture, Luke Hohmann, Addison wesley, 2003,
2. Software architecture, David M, Dikel, David Kane and James R, Wilson, Prentice Hall PTR, 2001
3. Software Design, David Budgen, second edition, Pearson education, 2003
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007,
5. Design Patterns in Java, Steven John Metsker & William C, Wake, Pearson education, 2006

6. J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003,
7. Design Patterns in C#, Steven John metsker, Pearson education, 2004,
8. Pattern Oriented Software Architecture, F,Buschmann &others, John Wiley & Sons,

Online Links:

https://nptel,ac,in/courses/106101061
http://www,digimat,in/nptel/courses/video/106105163/L01,html

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE PROJECT CONTROL & MONITORING

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25103a	PE-1	3	0	0	3	40	60	100

Course Objectives:

This course is designed to enable the students to understand the fundamental principles of Soft-ware Project management &will also have a good knowledge of the responsibilities of a project manager and how to handle them,

Course Outcomes: At the end of the course, Student will be able to

- CO1: Describe the fundamentals of Project Management
- CO2: Recognize and use Project Scheduling Techniques
- CO3: Familiarize with Project Control Mechanisms
- CO4: Understand Team Management
- CO5: Recognize the importance of Project Documentation and Evaluation

UNIT-I
Conventional Software Management: The water fall model, conventional software Management performance Evolution of Software Economics: software Economics, Pragmatic Software Cost Estimation Improving Software Economics: Reducing Software Product Size, Improving Software Processes, Improving Team Effectiveness, Improving Automation, Achieving Required Quality ,Peer Inspections,
UNIT-II
The old way and the new: The principles of convention al software Engineering, principles of modern software management, transitioning to an iterative process, Lifecycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases, Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts
UNIT-III
Work Flows of the process: Software process work flows, Inter Trans work flows, Check points of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments, Iterative Process Planning: work break down structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning
UNIT-1V
Process Automation: Automation Building Blocks, The Project Environment, Lecture9Hrs Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators Tailoring the Process: Process discriminants, Managing people and organizing teams,
UNIT-V
Project Organizations and Responsibilities: Line - of-Business Organizations, Project Organizations, evolution of Organizations, Future Software Project Management: modern Project Profiles, Next generation Software economics, modern process stransitions, Case Study: The Command Center Processing and Display System-Replacement(CCPDS-R)

Textbooks:

1. Software Project Management, Walker Royce, Pearson Education,2012
2. Bob Hughes, Mike Cotterell and RajibMall—Software Project Management, 6thEdition, Mc Graw Hill Edition, 2017

Reference Books:

1. PankajJalote,—SoftwareProjectManagementinpractice¦,5thEdition,PearsonEducation, 2017,
2. Murali K,Chemuturi,Thomas M,Cagley Jr,¦Mastering Software Project Management: Best Practices, Tools and Techniques¦, J,Ross Publishing, 2010
3. SanjayMohapatra,—SoftwareProjectManagement¦,CengageLearning,2011

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Online Links:

http://nptel,ac,in/courses/106101061/29
https://onlinecourses,nptel,ac,in/noc19_cs70/preview
https://nptel,ac,in/courses/110107430
https://nptel,ac,in/courses/110104073

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE METRICS AND QUALITY ENGINEERING

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25103b	PE-2	3	0	0	3	40	60	100

Course Objectives:

This course is designed to enable the students

1. To introduce the fundamental concepts of software quality, software process maturity, and quality frameworks,
2. To familiarize students with different types of software metrics and the importance of measure-ment in software engineering,
3. To develop the ability to apply software quality assurance practices, verification & validation, reviews, audits, and configuration management,

Course Outcomes: At the end of the course, Student will be able to

CO1: Understand software quality concepts, quality factors, and the impact of process maturity on software development,

CO2: Apply various software metrics to measure product, process, and project attributes effectively,

CO3: Calculate and interpret important product metrics such as LOC, Function Point, Halstead metrics, Cyclomatic Complexity, and Object-Oriented metrics (CK, MOOD, QMOOD),

CO4: Evaluate software processes using estimation metrics, defect metrics, testing metrics, and reliability models,

CO5: Apply software quality assurance activities, standards (ISO, CMMI), reviews, audits, SCM, and quality improvement models to enhance overall software quality,

UNIT-I: Introduction to Software Quality-
Definition of software quality- Quality concepts: Quality assurance, quality control, quality management- Software Engineering principles and role of quality- Software quality factors (McCall's, ISO 9126)- Defect Classification, Defect Density, Defect Removal Efficiency- Quality frameworks: CMM, CMMI, Six Sigma- Software Process Models & quality impact,
UNIT-II: Software Metrics
Basics- Need for measurement & metrics- Types of metrics: Product, Process, Project metrics- Measurement scales- Goal Question Metric (GQM)- Metrics validation & reliability
UNIT-III: Product & Process Metrics Product Metrics
Product & Process Metrics Product Metrics:- LOC metrics- Function Point metrics- Halstead metrics- Cyclomatic Complexity- OO Metrics: CK, MOOD, QMOOD Process Metrics:- Effort estimation - Productivity metrics- Defect metrics- Testing metrics- Maintenance metrics
UNIT-1V: Software Quality Assurance (SQA)
SQA activities & components- SQA standards: ISO 9001, ISO 9126, ISO 15504- Reviews & audits- SCM- Verification & Validation- Risk management- Reliability engineering,
UNIT-V: Quality Models & Testing Quality Models
McCall, Boehm, ISO 9126, Dromey- CMM & CMMI capability models Testing Quality:- TPI, TMMI- Defect tracking & lifecycle- RCA- Quality tools: Ishikawa, Pareto, checklists, control chart

Textbooks:

1. Jeff Tian: Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement Pearson Education,
2. Norman Fenton & James Bieman Software Metrics: A Rigorous and Practical Approach CRC Press,
3. Robert B, Grady: Practical Software Metrics for Project Management and Process Improvement Prentice Hall,

Reference Books:

1. Roger S, Pressman: **Software Engineering: A Practitioner's Approach** McGraw-Hill,
2. Ian Sommerville: **Software Engineering** Pearson Education,

3. Basu, S, K, & Phatak, Rajiv **Software Quality Assurance, Testing, and Metrics** PHI Learning,
4. Stephen H, Kan **Metrics and Models in Software Quality Engineering** Addison-Wesley,

Online Links:

https://appfire,com/resources/blog/software-development-quality-metrics
https://thesciencebrigade,com/jst/article/view/41
https://easychair,org/publications/preprint/XxM9

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ADVANCED SOFTWARE ENGINEERING

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25103c	PE-1	3	0	0	3	40	60	100

Course Objectives:

To provide students with a theoretical as well as practical understanding of agile software development practices and how small teams can apply them to create high-quality software,

- To provide good understanding of software design and a set of software technologies and APIs,
- To carry out detailed examination and demonstration of Agile development and testing techniques,
- To discuss Agile software development

Course Outcomes: At the end of the course, Student will be able to

CO1: Realize the importance of interacting with business stakeholders in determining the requirements for a software system

CO2: Perform iterative software development processes: how to plan them, how to execute them,

CO3: Point out the impact of social aspects on software development success,

CO4: Develop techniques and tools for improving team collaboration and software quality

CO5: Perform Software process improvement as an ongoing task for development teams,

UNIT-I: AGILE METHODOLOGY
Theories for Agile Management – Agile Software Development – Traditional Model vs, Agile Model - Classification of Agile Methods – Agile Manifesto and Principles – Agile Project Management – Agile Team Interactions – Ethics in Agile Teams - Agility in Design, Testing – Agile Documentations – Agile Drivers, Capabilities and Values
UNIT-II: AGILE PROCESSES
Lean Production - SCRUM, Crystal, Feature Driven Development- Adaptive Software Development - Extreme Programming: Method Overview – Lifecycle – Work Products, Roles and Practices,
UNIT-III: AGILITY AND KNOWLEDGE MANAGEMENT
Agile Information Systems – Agile Decision Making - Earl_S Schools of KM – Institutional Knowledge Evolution Cycle – Development, Acquisition, Refinement, Distribution, Deployment, Leveraging – KM in Software Engineering – Managing Software Knowledge – Challenges of Migrating to Agile Methodologies – Agile Knowledge Sharing – Role of Story-Cards – Story-Card Maturity Model (SMM),
UNIT-IV: AGILITY AND REQUIREMENTS ENGINEERING
Impact of Agile Processes in RE–Current Agile Practices – Variance – Overview of RE Using Agile Managing Unstable Requirements – Requirements Elicitation – Agile Requirements Abstraction Model – Requirements Management in Agile Environment, Agile Requirements Prioritization – Agile Requirements Modeling and Generation – Concurrency in Agile Requirements Generation,
UNIT-V: AGILITY AND QUALITY ASSURANCE
Agile Product Development – Agile Metrics – Feature Driven Development (FDD) – Financial and Production Metrics in FDD – Agile Approach to Quality Assurance - Test Driven Development – Agile Approach in Global Software Development,

Textbooks:

1. David J, Anderson and Eli Schragenheim, —Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results¦, Prentice Hall, 2003,

2. Hazza and Dubinsky, —Agile Software Engineering, Series: Undergraduate Topics in Computer Science¦, Springer, 2009,

Reference Books:

1. Craig Larman, —Agile and Iterative Development: A Manager_s Guide¦, Addison-Wesley, 2004,

2. Kevin C, Desouza, —Agile Information Systems: Conceptualization, Construction, and Management¦, Butterworth-Heinemann, 2007,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Online Links:

https://www,nptelvideos,com/video,php?id=904
http://www,digimat,in/nptel/courses/video/106101061/L26,html

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

AI-Driven Software Engineering

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25104a	PE-2	3	0	0	3	40	60	100

Course Objectives:

- To understand the foundational concepts of Artificial Intelligence and its role in modern software engineering,
- To learn how AI/ML/LLM techniques enhance various phases of the Software Development Life Cycle (SDLC),
- To introduce AI-driven approaches for requirements gathering, design automation, coding, testing, and maintenance,
- To enable students to use generative AI tools for improving development productivity and software quality,
- To analyze ethical, security, governance, and risk factors in AI-augmented software development,

Course Outcomes: At the end of the course, Student will be able to

CO1: Explain the fundamental concepts of AI-driven software engineering and its impact on modern SDLC,
CO2: Apply AI tools and NLP techniques for requirement analysis, modeling, and architectural design,
CO3: Use generative AI and intelligent automation tools for coding, refactoring, and DevOps processes,
CO4: Implement AI-based testing methods, defect prediction models, and quality engineering practices,
CO5: Evaluate ethical, governance, security, and risk considerations in AI-enabled software development,

UNIT-I: Introduction to AI-Driven Software Engineering
Evolution of SE Automation - AI, ML, DL & LLMs for SDLC - Generative AI & Autonomous Code Agents - Prompt Engineering Fundamentals - AI-Augmented SDLC Pipeline - Cognitive Automation & RPA - AI Tools Overview
UNIT-II: AI for Requirements Engineering
NLP in Requirements Elicitation - Requirement Classification & Prioritization - AI User Stories & Use Cases - Consistency Checking - Ambiguity Reduction - Requirements Validation
UNIT-III: AI-Driven Design & Architecture
AI-Assisted System Design - Automatic UML Diagram Generation - Pattern Recommendation - Model Driven Engineering with AI - Architecture Evaluation Metrics - Knowledge Graphs for Architecture,
UNIT-IV: AI-Powered Development & DevOps
LLM-Based Code Generation - Code Refactoring & Optimization - Secure Coding with AI - Intelligent CI/CD – AI Ops for Monitoring - Automated Code Review
UNIT-V: AI in Testing, Quality & Governance Testing:
Automated Test Case Generation - Bug Detection & Defect Prediction - AI Regression & Unit Testing Quality: - Reliability Prediction - Quality Metrics with ML Ethics & Governance: - AI Hallucination Risks - Responsible AI & IP Issues - Privacy, Security & Model Safety

Textbooks:

1. AI-Driven Software Development: A Modern Approach – Wiley Publications,
2. Generative AI for Software Engineering: Design, Development & Testing – Packt Publishing,
3. Machine Learning Applications in Software Engineering – Springer,

Reference Books:

1. Ian Sommerville, Software Engineering, Pearson Education,
2. Roger S, Pressman & Bruce R, Maxim, Software Engineering: A Practitioner's Approach, McGraw-Hill,
3. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press,

4. O'Reilly Media, Hands-On Large Language Models: Applications and Techniques,
5. Microsoft Research & GitHub, Applied AI for Developers (Open Learning Resource),

Online Links:

https://devin,ai/
https://www,mdpi,com/2076-3417/15/3/1344
https://www,preprints,org/manuscript/202504,0174

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CLOUD COMPUTING

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25104b	PE-2	3	0	0	3	40	60	100

Course Objectives:

- Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure, Compare the advantages and disadvantages of various cloud computing platforms,
- Investigate how a global storage solution can be optimized so that it can be delivered successfully from the cloud
- Evaluate information storage management design in a cloud environment and how it relates to the business objectives of an organization
- Analyze how best to provide reliable access to information both locally and remotely using storage technologies
- Critically appraise the opportunities and challenges of information management in complex business environments,

Course Outcomes: At the end of the course, Student will be able to

CO1: Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google App Engine,

CO2: Program data intensive parallel applications in the cloud,

CO3: Analyze the performance, scalability, and availability of the underlying cloud technologies and software,

CO4: Identify security and privacy issues in cloud computing,

CO5: Solve a real-world problem using cloud computing through group collaboration

UNIT-I
Definition, characteristics, components, Cloud service provider, the role of networks in Cloud computing, Cloud deployment models- private, public & hybrid, Cloud service models, multitenancy, Cloud economics and benefits, Cloud computing platforms - IaaS: Amazon EC2, PaaS: Google App Engine, Microsoft Azure, SaaS,
UNIT-II
Virtualization concepts, Server virtualization, Storage virtualization, Storage services, Network virtualization, Service virtualization, Virtualization management, Virtualization technologies and architectures, virtual machine, Measurement and profiling of virtualized applications, Hypervisors: KVM, Xen, VMware hypervisors and their features
UNIT-III
Relational databases, Cloud file systems: GFS and HDFS, Bigtable, HBase and Dynamo, MapReduce and extensions: Parallel computing, the map -Reduce model, Parallel efficiency of Map Reduce,
UNIT-IV
Advanced Cloud Concepts :Containers and Docker, Kubernetes and Cloud-native AI Workflows, Serverless Computing: AWS Lambda, Azure Functions, CI/CD Pipelines for AI Models in Cloud, Scaling AI Applications using Load Balancers and Auto-Scaling, Monitoring and Logging in Cloud for AI Workflows,
UNIT-V
Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud, Cloud computing security architecture: General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro - architectures; Identity Management and Access control, Autonomic security, Issues in cloud computing Implementing real time application over cloud platform, Issues in Inter - cloud environments, QOS Issues in Cloud, Dependability, data migration, streaming in Cloud, Quality of Service (QoS) monitoring in a Cloud computing environment, Cloud Middleware,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Textbooks:

1. Enterprise Cloud Computing by Gautam Shroff, Cambridge publication
2. Gautam Shroff, "Enterprise Cloud Computing Technology Architecture Applications", Cambridge University Press; 1 edition, [ISBN: 978-0521137355], 2010,
3. Rajkumar Buyya, Christian Vecchiola, S, Thamarai Selvi, Mastering Cloud Computing, McGraw-Hill,

Reference Books:

1. Cloud Security by Ronald Krutz and Russell Dean Vines, Wiley -India
2. Dr, Kumar Saurabh, "Cloud Computing", Wiley Publication
3. Dimitris N, Chorafas, "Cloud Computing Strategies" CRC Press; 1 edition [ISBN: 1439834539],2010
4. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach" McGraw Hill Osborne Media; 1 edition [ISBN: 0071626948], 200
5. RajkumarBuyya, James Broberg, Andrzej M, Goscinski, "Cloud Computing: Principles and Paradigms", Wiley Publication, 2011
6. Tim Mather, SubraKumaraswamy, Shahed Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'ReillyMediaInc, 2009

Online Links:

https://onlinecourses,nptel,ac,in/noc21_cs14/preview
https://nptel,ac,in/courses/106105167
https://onlinecourses,swayam2,ac,in/nou25_cs33/preview

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SECURE SOFTWARE DEVELOPMENT

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25104c	PE-2	3	0	0	3	40	60	100

Course Objectives:

- To understand fundamental security principles, software vulnerabilities, and the need for integrating security throughout the software development lifecycle,
- To learn and apply secure requirements engineering and architectural design practices, including threat modeling and risk assessment,
- To develop the ability to write secure code by following industry-recommended secure coding guidelines and preventing common software vulnerabilities,

Course Outcomes: At the end of the course, Student will be able to

CO1: fundamental security principles, common software vulnerabilities, and the importance of secure software development,

CO2: Apply security requirements engineering, threat modeling, and risk assessment techniques during software design,

CO3: Develop secure code using recommended secure coding practices to prevent common vulnerabilities such as SQL injection, XSS, and buffer overflows,

CO4: Evaluate software security using static analysis, dynamic analysis, penetration testing, and vulnerability assessment tools,

CO5: Implement secure deployment practices, DevSecOps pipelines, configuration hardening, patch management, and secure maintenance strategies,

UNIT-I: 1Fundamentals of software security
Fundamentals of software security, Need and importance of secure coding, Security principles: CIA triad, least privilege, defense in depth, Software vulnerabilities: causes and impacts, Secure Development Life Cycle (SDLC vs SSDLC), Security policies, standards, and guidelines,
UNIT-II: Requirements & Design Security
Secure Requirements Engineering: Functional vs Non-functional (Security) requirements, Risk analysis concepts, Security requirement tools and checklists, Secure Design: Secure architecture, Threat modeling (STRIDE, DREAD, PASTA),, Attack surface analysis, Design patterns for security ,Secure protocols and APIs, Misuse & abuse case
UNIT-III: Secure Coding Practices
Language-Specific Secure Coding: Common coding vulnerabilities (SQLi, XSS, CSRF, Buffer overflow, etc,), Secure coding guidelines (CERT, OWASP, MISRA), Input validation & error handling, Secure memory/resource management, Secure authentication & authorisation
UNIT-IV: Security Testing & Verification
Static Analysis: Code review, SAST tools, Dependency scanning, Dynamic Analysis: DAST, Fuzz testing, Penetration testing, Other Testing Approaches: Threat-based testing, Secure unit testing, Vulnerability scanning, Software composition analysis (SCA),
UNIT-V: Deployment & Maintenance Security
DevSecOps, Container security (Docker, Kubernetes), CI/CD pipeline security, Patch management, Secure configuration and hardening, Logging, monitoring, incident response, Secure release management,

Textbooks:

1. Gary McGraw, "Software Security: Building Security In", Addison-Wesley,
2. Ross Anderson, "Security Engineering: A Guide to Building Dependable Distributed Systems", Wiley,

3. Dafydd Stuttard & Marcus Pinto, "The Web Application Hacker's Handbook", Wiley,

Reference Books:

1. Mark Dowd, John McDonald, and Justin Schuh, "The Art of Software Security Assessment", Addison-Wesley,
2. Michael Howard & Steve Lipner, "The Security Development Lifecycle", Microsoft Press,
3. OWASP, "OWASP Secure Coding Practices – Quick Reference Guide",
4. NIST, "SP 800-64: Security Considerations in the System Development Life Cycle",

Online Links:

NIST Cyber security Resource Center – Standards, frameworks, and guidelines for SSDLC and software security, *Link:* https://csrc,nist,gov

https://www,coursera,org/learn/packt-secure-software-development-1o6mn

https://www,securecodewarrior,com/

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ADVANCED DATA STRUCTURES AND ALGORITHMS LAB

I M.Tech – I Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D58105	Professional Core	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	4	2	60	40	100

Course Objectives:

1. acquire practical skills in constructing and managing Data structures
2. apply the popular algorithm design methods in problem-solving scenarios

Course Outcomes: At the end of the course, Student will be able to

CO1: Design and develop programs to solve real world problems with the popular algorithm design methods,
CO2: Demonstrate an understanding of Non-Linear data structures by developing implementing the operations on AVL Trees, B-Trees, Heaps and Graphs,
CO3: Critically assess the design choices and implementation strategies of algorithms and data structures in complex applications
CO4: Utilize appropriate data structures and algorithms to optimize solutions for specific computational problems,
CO5: Compare the performance of different of algorithm design strategies,

List of Experiments:

1. **Implementation of Dictionary ADT using Arrays**
   Perform insert, delete, search and update operations,
2. **Dictionary Implementation using Linked Lists**
   Compare efficiency with array-based implementation,
3. **Hash Table Implementation with Separate Chaining**
   Resolve collisions using linked list chaining,
4. **Hash Table using Linear Probing**
   Implement open addressing with linear probing,
5. **Hash Table using Quadratic Probing**
   Handle collisions using quadratic probe sequence,
6. **Hash Table using Double Hashing**
   Implement and analyze second hash function usage,
7. **Rehashing Technique in Hash Tables**
   Dynamically resize hash table when load factor exceeds threshold,
8. **Extendible Hashing Implementation**
   Simulate directory-based dynamic hashing,
9. **Implementation of Skip List**
   Perform randomized insertion, deletion and searching,
10. **Probabilistic Analysis of Skip Lists**
    Measure search performance experimentally,
11. **Binary Search Tree (BST) Operations**
    Insert, delete and traverse nodes,
12. **AVL Tree Implementation**
    Implement rotations and balance factor maintenance,
13. **Red-Black Tree Bottom-Up Insertion**
    Demonstrate color fixing and rotations,
14. **Red-Black Tree Top-Down Deletion**
    Implement deletion algorithm with rebalancing,
15. **2-3 Tree Implementation**

Insert and search operations with node splitting,

16. **B-Tree Implementation**
Implement order-m B-tree with insertion and deletion,

17. **Comparative Study of BST, AVL and B-Tree Heights**
Analyze performance based on height and search time,

18. **Splay Tree Implementation**
Demonstrate splaying during access operations,

19. **Pattern Matching Algorithms Implementation**
Implement and compare:
- Brute Force
- Boyer-Moore
- KMP

20. **Text Processing Applications**
a) Huffman Coding for File Compression
b) Longest Common Subsequence (LCS) using Dynamic Programming
c) Construction of Suffix Tries and Compressed Tries

Textbooks:

1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition,
2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson- Freed, Silicon Press, 2008

Reference Books:

1. Fundamentals of Data Structures in C++, Horowitz Ellis, SahniSartaj, Mehta, Dinesh, 2ndEdition, Universities Press
2. Computer Algorithms/C++ Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2ndEdition, University Press
3. Data Structures and program design in C, Robert Kruse, Pearson Education Asia 4, An introduction to Data Structures with applications, Trembley & Sorenson, McGraw Hill

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS LABORATORY

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25106	Professional Core	0	0	4	2	60	40	100

Course Objectives:

- T o understand and implement **basic and advanced software design patterns**,
- To apply architectural styles and patterns in real-world software projects,
- To analyze software design problems and propose pattern-based solutions,
- To improve **software modularity, maintainability, and scalability** through proper design practices,
- To gain hands-on experience in **UML modeling** and pattern-based application development

Course Outcomes: At the end of the course, Student will be able to

CO1: Understand interrelationships, principles and guidelines governing architecture and evolution over time
CO2: Analyze the architecture and build the system from the components
CO3: Prepare creational patterns that deal with object creation mechanisms
CO4: Prepare structural patterns that ease the design by identifying a simple way to realize relationships among entities,
CO5: Learn behavioral patterns that identify common communication patterns between objects and realize these patterns

List of Experiments:

1. Presentation of the Project: Weather Mapping System,
2. Design of the Use Case View, Risk Analysis
3. Design of the Logical View of the Weather Mapping System (WMS),
4. Integration of selected architectural and design patterns in the logical view obtained previously
5. Design of the implementation, process, and deployment views for the Weather Mapping System
6. Generation from the previous architecture design of CORBA Interfaces and Components Definitions
7. Implementation of the Weather Mapping System (Java & C++), with a particular emphasis on the Inter process communication mechanism and the software components identified,
8. Use case Diagram for Librarian Scenario Using UML design Abstract factory design pattern
9. Using UML design Adapter-class Design pattern
   Using UML design Adapter-object Design pattern
10. Using UML design Strategy Design pattern
    Using UML design Builder Design pattern
11. Using UML design Bridge Design pattern
    Using UML design Decorator Design pattern

Reference Books:

- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley,
- Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Pearson,
- Martin Fowler, "Patterns of Enterprise Application Architecture", Addison-Wesley,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

RESEARCH METHODOLOGY AND IPR

(Common to M,Tech CSE, CN, SE,AI & ML)

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3DRM101		2	0	0	2	60	40	100

Course Objectives:

- Identify an appropriate research problem in their interesting domain,
- Understand ethical issues understand the Preparation of a research project thesis report,
- Understand the Preparation of a research project thesis report
- Understand the law of patent and copyrights,
- Understand the Adequate knowledge on IPR,

Course Outcomes (CO): Student will be able to

CO1:Analyze research related information

CO2:Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity,

CO3:Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular,

CO4:Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits,

UNIT - I
Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, scope, and objectives of research problem, Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations
UNIT - II
Effective literature studies approaches, analysis Plagiarism, Research ethics, Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee,
UNIT - III
Nature of Intellectual Property: Patents, Designs, Trade and Copyright, Process of Patenting and Development: technological research, innovation, patenting, development, International Scenario: International cooperation on Intellectual Property, Procedure for grants of patents, Patenting under PCT,
UNIT - IV
Patent Rights: Scope of Patent Rights, Licensing and transfer of technology, Patent information and databases, Geographical Indications,
UNIT - V
New Developments in IPR: Administration of Patent System, New developments in IPR; IPR of Biological Systems, Computer Software etc, Traditional knowledge Case Studies, IPR and IITs,

Text books:

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'"
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction",

Reference Books:

1. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners"

2. Halbert, “Resisting Intellectual Property”, Taylor & Francis Ltd ,2007,
3. Mayall, “Industrial Design”, McGraw Hill, 1992,
4. Niebel, “Product Design”, McGraw Hill, 1974,
5. Asimov, “Introduction to Design”, Prentice Hall, 1962,
6. Robert P, Merges, Peter S, Menell, Mark A, Lemley, “ Intellectual Property in New Technological Age”, 2016,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Software Process Automation Tools

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25107	Skill Enhancement Course	0	1	2	2	60	40	100

Course Objectives:

The course aims to:

- To provide **hands-on experience** with DevOps tools and process automation,
- To develop **skills in build, integration, testing, and deployment automation**,
- To familiarize students with **version control, CI/CD pipelines, and monitoring tools**,
- To enhance the ability to **analyze and implement DevOps workflows**,

Course Outcomes: At the end of the course, Student will be able to

CO1: Demonstrate version control using Git/GitHub/GitLab,
CO2: Implement automated builds using Maven, Gradle, or Ant,
CO3: Create and manage CI/CD pipelines using Jenkins or GitLab CI/CD,
CO4: Automate testing using Selenium, JUnit, or TestNG frameworks,
CO5: Deploy applications using Docker, Ansible, or other automation/configuration tools,

Module 1: Introduction to DevOps

- Agile development model overview
- DevOps concepts and ITIL framework
- DevOps lifecycle and Continuous Delivery
- Scrum, Kanban, and Release management
- Delivery pipeline and identifying bottlenecks

Experiments:

1. Explore Agile and DevOps lifecycle through a case study,
2. Create a sample Scrum board in Jira or Trello,
3. Identify bottlenecks in a sample delivery pipeline,

Module 2: Software Development Models & DevOps

Topics:

- DevOps lifecycle for business agility
- Continuous testing in DevOps
- Software architecture in DevOps: monolithic vs microservices
- Separation of concerns and database migrations
- Resilience in DevOps architecture

Experiments:

1. Compare monolithic vs microservices deployment with sample applications,
2. Implement a simple microservice with REST API,
3. Apply database migration using Flyway or Liquibase,

Module 3: Project Management for DevOps

Topics:

- Version control systems: need, history, and benefits
- Git basics: repositories, commits, branching, merging
- Code review tools: Gerrit, GitLab, pull requests
- Shared authentication and source control best practices

Experiments:

1. Initialize Git repository and perform basic operations (clone, commit, push, pull),
2. Create and merge branches using Git workflow,
3. Perform pull requests and code review using GitHub/GitLab,

Module 4: System Integration

Topics:

- Build automation tools: Maven, Gradle, Ant
- Jenkins build server setup, build dependencies, and plugins
- Build slaves and job chaining
- Pipeline design and Infrastructure as Code (IaC)

Experiments:

1. Create a Maven/Gradle project and automate build,
2. Setup Jenkins and create a basic CI job,
3. Chain multiple Jenkins jobs and configure build triggers,
4. Implement a simple IaC script using Ansible or Terraform,

Module 5: Testing & Deployment Automation

Topics:

- Types of testing in DevOps
- Automated testing frameworks: JUnit, TestNG, Selenium
- Test-driven development (TDD)
- Deployment systems: Docker, Puppet, Chef, Ansible, Kubernetes
- Monitoring CI/CD pipelines and analyzing reports

Experiments:

1. Write unit tests using JUnit/TestNG and integrate into Jenkins build,
2. Automate web testing using Selenium WebDriver,
3. Containerize a sample application using Docker,
4. Deploy the containerized application to Kubernetes,
5. Integrate SonarQube for code quality analysis,

Reference Books:

1. Jez Humble & David Farley, "Continuous Delivery", Addison-Wesley,
2. Joakim Verona, "Practical DevOps", Packt Publishing,
3. Deepak Gaikwad & Viral Thakkar, "DevOps Tools from Practitioner's Viewpoint", Wiley, 4, Len Bass, Ingo Weber & Liming Zhu, "DevOps: A Software Architect's Perspective", Addison-Wesley

St. Johns College of Engineering & Technology
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

AUDIT COURSE-I

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ENGLISH FOR RESEARCH PAPER WRITING

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3DAC101a	AC	2	0	0	0	60	40	100

Course Objectives: This course will enable students:

- Understand the essentials of writing skills and their level of readability
- Learn about what to write in each section
- Ensure qualitative presentation with linguistic accuracy

Course Outcomes (CO): Student will be able to

CO1: Understand the significance of writing skills and the level of readability
CO2: Analyze and write title, abstract, different sections in research paper
CO3: Develop the skills needed while writing a research paper

UNIT - I
1Overview of a Research Paper- Planning and Preparation- Word Order- Useful Phrases - Breaking up Long Sentences-Structuring Paragraphs and Sentences-Being Concise and Removing Redundancy -Avoiding Ambiguity
UNIT - II
Essential Components of a Research Paper- Abstracts- Building Hypothesis-Research Problem - Highlight Findings- Hedging and Criticizing, Paraphrasing and Plagiarism, Cauterization
UNIT - III
Introducing Review of the Literature – Methodology - Analysis of the Data-Findings - Discussion-Conclusions-Recommendations,
UNIT - IV
Key skills needed for writing a Title, Abstract, and Introduction
UNIT - V
Appropriate language to formulate Methodology, incorporate Results, put forth Arguments and draw Conclusions

Suggested Reading:

1. Goldbort R (2006) Writing for Science, Yale University Press (available on Google Books) Model Curriculum of Engineering & Technology PG Courses [Volume-I]
2. Day R (2006) How to Write and Publish a Scientific Paper, Cambridge University Press
3. Highman N (1998), Handbook of Writing for the Mathematical Sciences, SIAM, Highman'sbook
4. Adrian Wallwork , English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DISASTER MANAGEMENT

I M, Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3DAC101b	AC	2	0	0	0	60	40	100

Course Objectives: This course will enable students:

- Learn to demonstrate critical understanding of key concepts in disaster risk reduction and hu- manitarian response,
- Critically evaluate disaster risk reduction and humanitarian response policy and practice from multi- ple perspectives,
- Develop an understanding of standards of human Italian response and practical relevance in specific types of disasters and conflict situations
- Critically understand the strengths and weaknesses of disaster management approaches, planningand programming in different countries, particularly their home country or the countries they work in

UNIT - I

Introduction: Disaster: Definition, Factors and Significance; Difference Between Hazard and Disaster; Natural and Manmade Disasters: Difference, Nature, Types and Magnitude,
Disaster Prone Areas in India:
Study of Seismic Zones; Areas Prone to Floods and Droughts, Landslides and Avalanches; Areas Prone to Cyclonic and Coastal Hazards with Special Reference to Tsunami; Post- Disaster Diseases and Epidemics

UNIT - II

Repercussions of Disasters and Hazards:
Economic Damage, Loss of Human and Animal Life, Destruction of Ecosystem, Natural Disasters Earthquakes, Volcanisms, Cyclones, Tsunamis, Floods, Droughts and Famines, Landslides and Avalanches, Man-made disaster: Nuclear Reactor Meltdown, Industrial Accidents, Oil Slicks and Spills, Outbreaks of Disease and Epidemics, War and Conflicts,

UNIT - III

Disaster Preparedness and Management:
Preparedness: Monitoring of Phenomena Triggering A Disasteror Hazard; Evaluation of Risk: Application of Remote Sensing, Data from Meteorological and Other Agencies, Media Reports: Governmental and Community Preparedness,

UNIT - IV

Risk Assessment Disaster Risk:
Concept and Elements, Disaster Risk Reduction, Global and National Disaster Risk Situation,
Techniques of Risk Assessment, Global Co-Operation in Risk Assessment and Warning, People's Participation in Risk Assessment, Strategies for Survival,

UNIT - V

Disaster Mitigation:
Meaning, Concept and Strategies of Disaster Mitigation, Emerging Trends In Mitigation, Structural Mitigation and Non-Structural Mitigation, Programs of Disaster Mitigation in India,

Suggested Reading

1. R,Nishith, Singh A K, "Disaster Management in India: Perspectives, issues and strategies
2. "'New Royal book Company, ,Sahni, Pardeep Et,Al,(Eds,), "Disaster Mitigation Experiences And Reflec- tions", Prentice Hall Of India, New Delhi,
3. Goel S,L,, Disaster Administration And Management Text And Case Studies", Deep & Deep

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SANSKRIT FOR TECHNICAL KNOWLEDGE

I M.Tech – I Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
25G3DAC101c	AC	L	T	P	C	CIA	SEE	Total
		2	0	0	0	60	40	100

Course Objectives: This course will enable students:

- To get a working knowledge in illustrious Sanskrit, the scientific language in the world
- Learning of Sanskrit to improve brain functioning
- LearningofSanskrittodevelopthelogicinmathematics,science&othersubjects enhancing the memory power
- The engineering scholars equipped with Sanskrit will be able to explore the huge Knowledge from ancient literature

Course Outcomes (CO): Student will be able to CO1: Understanding basic Sanskrit language

CO2: Ancient Sanskrit literature about science &technology can be understood

CO3: Being a logical language will help to develop logic in students

UNIT - I
Alphabets in Sanskrit,
UNIT - II
Past/Present/Future Tense, Simple Sentences
UNIT - III
Order, Introduction of roots
UNIT - IV
Technical information about Sanskrit Literature
UNIT - V
Technical concepts of Engineering-Electrical, Mechanical, Architecture, Mathematics

Suggested Reading

1. "Abhyaspustakam" –Dr,Vishwas, Sanskrit-Bharti Publication, New Delhi
2. "Teach Yourself Sanskrit" Prathama Deeksha-VempatiKutumbshastri, RashtriyaSanskrit Sansthanam, New Delhi Publication
3. "India's Glorious ScientificTradition" Suresh Soni, Ocean books (P) Ltd,,New Delhi

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Software Testing and Verification

I M.Tech – II Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
25G3D25201	Engineering Science	3	0	0	3	40	60	100

Course Objectives:

This course is designed to enable the students

1. Understand the role of **software testing and verification** in the software development life cycle (SDLC),
2. Analyze and apply various **software testing methodologies and techniques**,
3. Prepare test plans, test cases, and implement those using **testing tools**,
4. Use testing automation tools for performance, regression, and system testing,
5. Evaluate and improve **quality assurance processes** and test management strategies,

Course Outcomes: A student after completion of the course will be able to

CO1: Explain software testing fundamentals and verification vs validation,
CO2: Apply black box, white box, and advanced testing techniques,
CO3: Develop test plans, test cases, and manage testing cycles,
CO4: Use industry tools for test automation and performance testing,
CO5: Critically evaluate testing results and improve quality management processes,

UNIT-1
Introduction to Software Testing and Verification: Verification vs Validation (V & V),Role of Testing in SDLC,Software Quality Assurance (SQA) Fundamentals, Test Process and Test Life Cycle, Levels of Testing (Unit, Integration, System, Acceptance),Test Planning & Documentation, Test Policy and Test Strategy, Metrics for Software Testing,
UNIT-1I
Software Testing Techniques: Black Box Testing (Equivalence, Boundary Value), White Box Testing (Statement, Branch, Path), Structural & Functional Testing, Regression Testing, Risk-Based Testing, Fault and Defect Classification, Mutation Testing, Model Based Testing, Exploratory Testing,
UNIT-III
Test Planning & Test Case Design: Creating Test Plans, Test Case Design Techniques, Test Data Preparation, Test Coverage Criteria, Traceability Matrix, Use Case Testing, Prioritization and Test optimization, Defect Lifecycle and Tracking, Test Reporting & Metrics,
UNIT-IV
Testing Tools and Automation: Test Tool Taxonomy and Selection Criteria, Automation Frameworks,LoadRunner / JMeter for Performance Testing, Selenium / JUnit for Automation,WinRunner / Rational Tools, Java Testing Tools (JUnit, TestNG),Test Script Recording & Playback, Continuous Integration & Automated Testing,Introduction to Modern CI/CD Testing Tools,
UNIT-V
Advanced Testing & Special Topics: Acceptance & Operational Testing, Testing Client/Server & Web Applications, Security and Performance Testing, Testing of Object-oriented Systems, Mobile Application Testing, Test Metrics & Test Effectiveness Evaluation, Post-Implementation Review, Test Management Challenges, Industry Case Studies & Best Practices,

Text books:

1. **Effective Methods for Software Testing**, William E, Perry, Wiley India,
2. **Software Testing – Principles and Practices**, Naresh Chauhan, Oxford University Press,
3. **Software Quality Assurance – From Theory to Implementation**, Daniel Galin, Pearson Education,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books:

1. **Testing Computer Software**, Cem Kaner, Jack Falk, Hung Quoc Nguyen, Wiley India,
2. **Software Testing Techniques**, Boris Beizer, Dreamtech Press,
3. **The Craft of Software Testing**, Brian Marick, Pearson Education,
4. **Foundations of Software Testing**, Aditya P, Mathur, Pearson,
5. **Software Testing in the Real World**, Edward Kit, Pearson,

Online Links:

1. https://www,tutorialspoint,com/advanced_data_structures/index,asp
2. http://peterindia,net/Algorithms,html 3, Abdul Bari,1, Introduction to Algorithms (youtube,com)

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Advanced Algorithms and Optimization Techniques

I M.Tech – II Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25202	Engineering Science	3	0	0	3	40	60	100

Course Objectives:

This course is designed to enable the students

1. Understand advanced algorithm design paradigms and complexity analysis,
2. Formulate and model optimization problems from real-world scenarios,
3. Apply classical, numerical, and modern computational optimization techniques,
4. Analyze performance and correctness of algorithms and optimization methods,
5. Use algorithmic and optimization tools to solve large and complex problems,

Course Outcomes: A student after completion of the course will be able to

CO1: Apply advanced algorithmic strategies (e,g,, divide-and-conquer, dynamic programming, greedy) to solve complex computational problems,
CO2: Evaluate algorithm efficiency using complexity and performance metrics,
CO3: Formulate optimization problems and select appropriate solution techniques,
CO4: Implement classical and heuristic optimization algorithms (e,g,, simplex, GA, local search),
CO5: Integrate algorithmic and optimization methods for real-world engineering and data problems,

UNIT-1
Advanced Algorithm Design Techniques: Divide-and-Conquer strategies, Dynamic Programming (DP),Greedy Algorithms, Branch and Bound, Backtracking Methods, Randomized Algorithms, Approximation Algorithms, Duality & Linear Programming basics, Complexity Classes and NP-Completeness principles,
UNIT-1I
Graph & Search Algorithms: Shortest Path Algorithms (Dijkstra, Bellman-Ford),Minimum Spanning Trees (Kruskal/Prim),Network Flows (Ford-Fulkerson, Edmonds-Karp),Matching & Cover Problems, A* Search and Heuristics, Constraint Programming & Search Space Pruning, Graph Coloring & Partitioning, Spectral Graph Algorithms basics, Advanced DFS/BFS Applications,
UNIT-III
Classical Optimization Techniques: Introduction to Optimization — problem formulation, Single variable optimization, Multivariable optimization (with/without constraints),Lagrange Multipliers & Kuhn–Tucker conditions, Convex sets and convex optimization fundamentals, Linear Programming — Simplex (Two-Phase/Big-M),Duality Theory in LP,Integer Programming — Branch & Bound, Cutting Planes, Sensitivity & Parametric Analysis
UNIT-IV
Numerical and Metaheuristic Optimization: Gradient-based methods (Steepest descent, Newton),Conjugate Gradient and Quasi-Newton methods, elder–Mead Simplex Search, Penalty & Barrier Methods, Genetic Algorithms (GA) — crossover, mutation,Multi-Objective GA and Pareto Optimization, Particle Swarm Optimization (PSO), Differential Evolution, Simulated Annealing, Local Search & Tabu Search heuristic methods,
Applications & Advanced Topics: Optimization in Machine Learning (loss minimization, gradient descent),Algorithmic applications in scheduling and routing, Convex vs non-convex problems, Distributed & Parallel optimization methods, Online & stochastic optimization, Real-world case studies (network design, resource allocation),Constraint optimization in AI/ML,Performance benchmarking and trade-offs, Software tools for algorithms & optimization (e,g,, MATLAB, Python libraries),

Text books:

1. **Introduction to Algorithms**, Cormen, Leiserson, Rivest, Stein — comprehensive algorithm design,
2. **The Design of Approximation Algorithms**, David P, Williamson & David B, Shmoys — approximation and optimization,
3. **Convex Optimization**, Stephen Boyd & Lieven Vandenberghe — mathematical optimization techniques,

4. **Genetic Algorithms in Search, Optimization, and Machine Learning**, David E, Goldberg — evolutionary computing,

Reference Books:

1. **Advanced Algorithms and Data Structures**, various university lecture notes
2. **Numerical Optimization**, Jorge Nocedal & Stephen Wright — detailed optimization methods
3. **Optimization Theory with Applications**, S,S, Rao
4. Research papers and tutorials on specific heuristic/metaheuristic methods,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Intelligent Software Systems

I M.Tech – II Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25203a	PE-3	3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand the foundational principles of intelligent systems and software,
2. Analyze how software systems can exhibit "intelligent" behavior via AI techniques,
3. Apply search, reasoning, and learning algorithms to realistic problems,
4. Design and evaluate intelligent agents using representation and decision techniques,
5. Integrate intelligent components into software systems to solve complex tasks,

Course Outcomes: A student after completion of the course will be able to

CO1: Define the scope and limits of intelligent software systems,
CO2: Apply search and optimization methods for problem solving,
CO3: Use knowledge representation and reasoning techniques,
CO4: Implement learning-based algorithms for adaptive behavior,
CO5: Engineer intelligent components within software applications,

UNIT-1
Introduction to Intelligent Software Systems: What are Intelligent Systems? Intelligent Software vs Traditional Software, Intelligent Agents and Environments, Rationality and Autonomy in Software, Software Architectures Supporting Intelligence, Overview of AI Techniques for Software, Case Studies of Intelligent Software, Ethical and Practical Challenges, Intelligent System Lifecycle,
UNIT-1I
Search & Optimization Methods: Problem solving & State Space, Uninformed Search (BFS, DFS),Informed Search (A*, Greedy Best-First),Local Search & Hill Climbing, Optimization Algorithms, Heuristics and Evaluation Functions, Constraint Satisfaction Problems, Search in Software Decision Making, Search Performance Metrics,
UNIT-III
Knowledge Representation & Reasoning: Logical Representation (Propositional & FOL),Rule-based Systems & Expert Systems, Semantic Networks & Ontologies, Inference Engines & Reasoning, Uncertainty Handling (Bayesian methods),Frames and Production Rules, Case-based Reasoning, Knowledge Engineering Practices, Software Design Using Knowledge Models,
UNIT-IV
Learning & Adaptive Techniques: Machine Learning Overview, Supervised Learning for Software Behavior, Unsupervised Learning & Clustering, Reinforcement Learning in Software Agents, Neural Networks & Deep Learning Basics, Genetic Algorithms & Evolutionary Computing, Statistical Learning Approaches, Feature Extraction and Selection, Integrating Learning into Software,
UNIT-V
Intelligent Software Applications & Trends: Intelligent Decision Support Systems, Intelligent Human-Computer Interfaces, Software Agents & Multi-Agent Systems, Natural Language Interfaces, Intelligent Software in IoT/Cloud, Autonomous Software Systems, Explainable & Ethical AI in Software,Software Testing for Intelligent Systems, Future Research Directions,

Text books:

1,Artificial Intelligence: A Modern Approach — Stuart Russell & Peter Norvig (Pearson), Overall foundation for intelligent systems methods,
2,Intelligent Systems: Architecture, Design, and Control — Meystel & Albus, Focuses on design of intelligent systems & agents,
3,Soft Computing and Intelligent Systems Design — Karray & De Silva, Shows hybrid approaches in intelligent systems,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books:

1. **Knowledge Representation & Reasoning** — Ronald Brachman & Hector Levesque,
2. **Machine Learning** — Tom M, Mitchell,
3. **Reinforcement Learning: An Introduction** — Sutton & Barto,
4. **Expert Systems: Principles and Programming** — Giarratano & Riley,
5. **Intelligent Agents: Theory and Practice** — various academic collections,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Enterprise Application Development

I M.Tech – II Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D25203b	PE3	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand the concepts and architecture of **enterprise-scale software systems**,
2. Learn **multi-tier, distributed, and component-based application development**,
3. Design and develop **secure, scalable, and maintainable enterprise applications**,
4. Apply modern frameworks and technologies for **web and enterprise development**,
5. Gain practical knowledge of **enterprise integration, deployment, and testing**,

Course Outcomes: A student after completion of the course will be able to

CO1: Explain enterprise application architectures and design principles,
CO2: Design multi-tier and distributed enterprise applications,
CO3: Develop enterprise applications using Java/,NET frameworks,
CO4: Implement security, transaction management, and persistence mechanisms,
CO5: Deploy, test, and maintain enterprise applications in real-world environments,

UNIT-1
Introduction to Enterprise Application Development: Enterprise Applications – Definition and Characteristics, Evolution of Enterprise Systems, Challenges in Enterprise Application Development, Enterprise Application Architecture,Multi-Tier Architecture (Presentation, Business, Data layers),Client-Server vs Distributed Systems, Component-Based Software Development, Enterprise Application Lifecycle, Case Studies of Enterprise Applications,
UNIT-1I
Enterprise Architecture & Design Patterns: Layered Architecture,Service-Oriented Architecture (SOA),Micro services Architecture – Overview,Model-View-Controller (MVC) Pattern,Enterprise Design Patterns,Data Access Patterns,Dependency Injection and Inversion of Control,Object-Relational Mapping (ORM) Concepts,Architectural Best Practices,
UNIT-III
Java Enterprise Application Development: Java EE / Jakarta EE Architecture,Servlets and Java Server Pages (JSP),JavaBeans and Enterprise JavaBeans (EJB),Session Management,Java Persistence API (JPA),Transactions and Concurrency, Web Services (SOAP & REST),Exception Handling in Enterprise Applications,Enterprise Application Packaging and Deployment,
UNIT-IV
Enterprise Integration, Security & Transactions: Enterprise Integration Patterns, Messaging Systems (JMS, Message Queues),Web Services Integration, Security Requirements in Enterprise Systems, Authentication and Authorization, Role-Based Access Control, Secure Communication (SSL, HTTPS),Transaction Management (ACID Properties),Distributed Transactions,
UNIT-V
Modern Enterprise Applications & Deployment: Spring Framework Overview, Spring MVC and Spring Boot,Microservices using Spring, Cloud-Based Enterprise Applications, Containerization and Dockers Basics,DevOps and CI/CD for Enterprise Systems, Testing Enterprise Applications, Performance and Scalability Issues, Industry Case Studies and Best Practices

Text books:

1. **Enterprise Application Architecture** – Martin Fowler, Addison-Wesley
2. **Java EE 8 Application Development** – David R, Heffelfinger, Packt
3. **Spring in Action** – Craig Walls, Manning Publications

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books:

1. **Designing Enterprise Applications with Java EE** – Bill Burke
2. **Core J2EE Patterns** – Alur, Crupi, Malks
3. **Patterns of Enterprise Application Architecture** – Martin Fowler
4. **Professional Java EE Design Patterns** – Packt Publishing
5. **Microservices Patterns** – Chris Richardson

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Model-Driven Software Engineering

I M.Tech – II Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25203c	PE-3	3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand the principles and motivation behind **Model-Driven Software Engineering (MDSE)**,
2. Learn different **modeling languages and meta-modeling concepts**,
3. Apply **model transformation and code generation techniques**,
4. Analyze the role of models in improving software quality and productivity,
5. Design and develop software systems using **model-centric approaches**,

Course Outcomes: A student after completion of the course will be able to

CO1: Explain the concepts, benefits, and challenges of MDSE,
CO2: Develop system models using UML and domain-specific languages,
CO3: Apply meta-modeling and model transformation techniques,
CO4: Generate executable code and documentation from models,
CO5: Evaluate model-driven approaches in real-world software projects,

UNIT-1
Introduction to Model-Driven Software Engineering: Evolution of Software Engineering Paradigms, Model-Centric vs Code-Centric Development, Principles of Model-Driven Engineering (MDE),Model-Driven Architecture (MDA) – OMG Standard, Platform Independent Model (PIM),Platform Specific Model (PSM),Computation Independent Model (CIM),Benefits and Limitations of MDSE,Case Studies in Model-Driven Development,
UNIT-1I
Modeling Languages and UML: Introduction to Software Modeling, Unified Modeling Language (UML) Overview, Structural Modeling (Class, Object Diagrams),Behavioral Modeling (Sequence, Activity Diagrams),Architectural Modeling (Component, Deployment Diagrams),UML Profiles and Extensions, Model Consistency and Validation, Modeling Best Practices, Tool Support for UML Modeling,
UNIT-III
Meta-Modeling and Domain-Specific Modeling: Concepts of Meta-Modeling,Meta-Object Facility (MOF),Abstract Syntax and Concrete Syntax,Domain-Specific Languages (DSLs),Designing DSLs,Textual vs Graphical DSLs,Constraints and Semantics in DSLs,Model Validation and Constraints (OCL),Case Studies of DSL-Based Systems,
UNIT-IV
Model Transformation and Code Generation: Model-to-Model (M2M) Transformations, Model-to-Text (M2T) Transformations, Transformation Languages (ATL, QVT),Code Generation Techniques, Templates and Transformation Rules, Synchronization and Model Consistency, Reverse Engineering and Round-Trip Engineering, Tool Support for Model Transformation, Testing and Debugging Model Transformations,
UNIT-V
Advanced Topics and Applications: Model-Driven Development Tools (Eclipse Modeling Framework – EMF),Software Product Lines and Variability Modeling, Model-Driven Testing, Model-Based Verification and Validation,MDSE in Agile Development, Industrial Applications of MDSE,Challenges and Research Issues in MDSE,Integration with DevOps and CI/CD,Future Trends in Model-Driven Engineering,

Text books:

1. **Model-Driven Software Engineering in Practice** – Marco Brambilla, Jordi Cabot, Manuel Wimmer, Morgan Kaufmann,
2. **Model-Driven Architecture: Applying MDA to Enterprise Computing** – David S, Frankel, Wiley,
3. **Domain-Specific Modeling: Enabling Full Code Generation** – Steven Kelly & Juha-Pekka Tolvanen, Wiley,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books:

1. **Software Abstractions: Logic, Language, and Analysis** – Daniel Jackson,
2. **UML Distilled** – Martin Fowler,
3. **Metamodeling and Model-Driven Engineering** – various academic lecture notes,
4. **Principles of Model-Driven Software Engineering** – Jörg Kienzle et al,
5. Research papers from **IEEE Software**, **ACM TOSEM**, **SoSyM Journal**,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Machine Learning in Software Engineering Processes

I M.Tech – II Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D25204a	PE-4	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand the role of **Machine Learning (ML)** in improving software engineering processes,
2. Learn how ML techniques can support **requirements, design, testing, and maintenance**,
3. Apply ML models to **predict, automate, and optimize software development activities**,
4. Analyze real-world software engineering data using ML techniques,
5. Evaluate challenges, ethics, and future directions of ML-driven software engineering,

Course Outcomes: A student after completion of the course will be able to

CO1: Explain how machine learning can be integrated into software engineering processes,
CO2: Apply ML techniques to software analytics and process improvement,
CO3: Build predictive models for defects, effort, and reliability estimation,
CO4: Use ML methods to support software testing and maintenance activities,
CO5: Evaluate ML-based solutions for software engineering problems in industry settings,

UNIT-1
Foundations of ML for Software Engineering: Introduction to Machine Learning, Software Engineering Process Models, Software Engineering Data and Metrics, Data Collection in Software Projects, Supervised vs Unsupervised Learning, Feature Engineering for Software Data,ML Pipelines for SE Applications, Evaluation Metrics for ML Models, Challenges of Applying ML in SE
UNIT-1I
ML in Requirements & Design: Requirements Engineering Overview, Text Mining for Requirements Analysis, Natural Language Processing for SRS,Requirements Classification and Clustering, Traceability Using ML,Design Pattern Detection Using ML,Architecture Analysis with Learning Models, Recommender Systems for Design Decisions, Case Studies in Requirements & Design Automation,
UNIT-III
ML in Software Testing: Software Testing Lifecycle, Test Case Prioritization Using ML,Test Case Generation Techniques, Fault Prediction and Defect Classification, Regression Testing Optimization, Test Effort Estimation, Test Oracle Automation, Coverage Prediction Models,ML Tools for Software Testing,
UNIT-IV
ML in Maintenance & Evolution: Software Maintenance Processes, Bug Report Analysis Using ML,Defect Prediction Models, Code Smell Detection, Change Impact Analysis, Software Refactoring Recommendations, Effort Estimation Using ML,Predictive Maintenance Models, Software Evolution Analytics,
UNIT-V
Advanced Topics & Industrial Applications: Mining Software Repositories (MSR),Deep Learning in Software Engineering, Reinforcement Learning for SE Automation, DevOps Analytics and AIOps,Explainable AI in Software Engineering, Ethical Issues and Bias in ML Models, Tool Support and Frameworks, Industry Case Studies (Google, Microsoft),Research Trends and Future Directions,

Text books:

1. **Machine Learning for Software Engineering** – Tim Menzies, Tao Xie, David Lo,
2. **Data Mining for Software Engineering** – Mark Harman, ACM Publications,
3. **Introduction to Machine Learning** – Ethem Alpaydin, MIT Press,

Reference Books:

1. **Mining Software Repositories** – Hassan & Xie, Springer,
2. **Software Analytics** – Ahmed E, Hassan, Morgan Kaufmann,
3. **Applied Machine Learning** – Kevin P, Murphy,
4. **Pattern Recognition and Machine Learning** – Christopher M, Bishop,
5. IEEE & ACM research papers on ML in SE,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Distributed Systems

I M.Tech – II Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
25G3D25204b	PE-4	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand the principles, architectures, and models of **distributed systems**,
2. Learn communication, synchronization, and coordination mechanisms,
3. Study distributed algorithms for **consistency, fault tolerance, and consensus**,
4. Analyze distributed storage, naming, and resource management techniques,
5. Design scalable and reliable distributed applications,

Course Outcomes: A student after completion of the course will be able to

CO1: Explain distributed system concepts, goals, and design challenges,
CO2: Apply communication and synchronization techniques in distributed environments,
CO3: Analyze distributed algorithms for consistency and coordination,
CO4: Design fault-tolerant and reliable distributed systems,
CO5: Evaluate modern distributed system architectures and applications,

UNIT-1
Introduction to Distributed Systems: Definition and Characteristics of Distributed Systems, Goals: Transparency, Scalability, Reliability, Types of Distributed Systems, Distributed System Architectures, Hardware and Software Models, Communication Models, Design Issues and Challenges, Distributed System Examples, Case Studies (Cluster, Cloud, P2P),
UNIT-1I
Communication & Synchronization: Inter-Process Communication (IPC) ,Remote Procedure Call (RPC),Remote Method Invocation (RMI),Message-Oriented Communication, Logical Clocks (Lamport, Vector Clocks),Clock Synchronization Algorithms, Mutual Exclusion in Distributed Systems, Election Algorithms, Distributed Deadlocks,
UNIT-III
Distributed Algorithms & Consistency: Distributed Transactions, Two-Phase and Three-Phase Commit Protocols, Consistency Models (Strict, Sequential, Eventual),Replication Techniques, Distributed Shared Memory, Consensus Algorithms (Paxos, Raft),Atomic Broadcast, Concurrency Control, CAP Theorem,
UNIT-IV
Fault Tolerance & Security: Fault Models in Distributed Systems, Failure Detection,Checkpointing and Recovery, Replication for Fault Tolerance, Byzantine Fault Tolerance, Distributed Security Issues, Authentication and Authorization, Secure Communication, Intrusion Detection in Distributed Systems,
UNIT-V
Distributed File Systems & Modern Systems: Distributed File Systems (NFS, AFS),Distributed Naming Services, Cloud Computing and Distributed Systems, Distributed Databases, Big Data Processing Systems (Hadoop, Spark),Micro services Architecture, Containerization and Orchestration, Peer-to-Peer Systems,Case Studies (Google File System, HDFS),

Text books:

1. **Distributed Systems: Concepts and Design** – George Coulouris, Jean Dollimore, Tim Kindberg, Pearson,
2. **Distributed Systems** – Maarten van Steen & Andrew S, Tanenbaum, Pearson,
3. **Principles of Distributed Database Systems** – M, Tamer Özsu & Patrick Valduriez,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books:

1. **Designing Data-Intensive Applications** – Martin Kleppmann,
2. **Distributed Algorithms** – Nancy Lynch,
3. **Reliable Distributed Systems** – Kenneth P, Birman,
4. **Modern Operating Systems** – Andrew S, Tanenbaum (Distributed OS parts),
5. IEEE & ACM research papers on distributed systems,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DevOps and Continuous Engineering

I M.Tech – II Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D25204c	PE-4	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand DevOps culture, principles, and practices in modern software engineering,
2. Learn continuous integration, delivery, and deployment pipelines,
3. Apply automation tools for build, test, release, and monitoring,
4. Design scalable, reliable, and secure DevOps workflows,
5. Analyze continuous engineering practices in real-world environments,

Course Outcomes: A student after completion of the course will be able to

CO1: Explain DevOps concepts and continuous engineering lifecycle,
CO2: Design and implement CI/CD pipelines using DevOps tools,
CO3: Automate testing, configuration, and deployment processes,
CO4: Apply monitoring, logging, and reliability engineering practices,
CO5: Evaluate DevOps practices for scalability, security, and performance,

UNIT-1
Introduction to DevOps & Continuous Engineering: Evolution of Software Development Practices, DevOps Culture and Principles, Continuous Engineering Concepts, DevOps Lifecycle, Agile vs DevOps, DevOps Tool chain Overview, Infrastructure as Code (IaC) Basics,DevOps Metrics and KPIs,Industry Case Studies,
UNIT-1I
Continuous Integration & Continuous Delivery: Continuous Integration (CI) Fundamentals, Build Automation Tools, Version Control Systems (Git),CI Tools (Jenkins, GitHub Actions),Automated Unit and Integration Testing, Continuous Delivery vs Continuous Deployment, Artifact Management, Branching and Release Strategies, I/CD Pipeline Design
UNIT-III
Configuration Management & Infrastructure Automation: Configuration Management Concepts, Infrastructure as Code (IaC),Configuration Tools (Ansible, Puppet, Chef),Containerization Fundamentals,Docker Architecture and Usage, Container Orchestration (Kubernetes),Cloud Platforms for DevOps, Deployment Strategies (Blue-Green, Canary),Security in Configuration Management,
UNIT-IV
Continuous Testing, Monitoring & Reliability: Test Automation Strategies, Test Types in DevOps, Continuous Testing Tools, Monitoring and Logging Basics, Observability in DevOps, Application Performance Monitoring (APM),Reliability Engineering (SRE) Concepts, Incident Management and Root Cause Analysis, Feedback Loops and Continuous Improvement,
UNIT-V
Advanced DevOps Practices & Future Trends: DevSecOps – Integrating Security, DevOps for Microservices,Cloud-Native DevOps, Continuous Engineering for AI/ML Systems (MLOps),Scaling DevOps Practices, Compliance and Governance,Toolchain Integration, Case Studies (Netflix, Amazon, Google),Emerging Trends in DevOps,

Text books:

1. **Continuous Delivery** – Jez Humble & David Farley, Addison-Wesley,
2. **The DevOps Handbook** – Gene Kim, Jez Humble, Patrick Debois, John Willis,
3. **Site Reliability Engineering** – Betsy Beyer et al,, O'Reilly,

Reference Books:

1. **DevOps: A Software Architect's Perspective** – Len Bass, Ingo Weber, Liming Zhu,
2. **Infrastructure as Code** – Kief Morris,
3. **Kubernetes Up & Running** – Kelsey Hightower,
4. **Accelerate** – Nicole Forsgren, Jez Humble, Gene Kim,
5. Research papers from IEEE & ACM on DevOps,

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Software Testing and Verification Laboratory

I M.Tech – II Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25201P	PE-4	3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Provide hands-on experience with **software testing and verification techniques**,
2. Enable students to design **test plans, test cases, and test scripts**,
3. Apply **black box and white box testing methods** practically,
4. Use **automated testing tools** for functional and performance testing,
5. Understand **defect tracking, reporting, and test management** practices,

Course Outcomes: A student after completion of the course will be able to

CO1: Design and execute test cases using systematic testing techniques,
CO2: Apply white box and black box testing on software programs,
CO3: Perform regression and integration testing,
CO4: Use automated testing tools for functional and performance testing,
CO5: Analyze test results and prepare professional test reports,

List of Experiments

Experiment 1

Study of Software Testing Process, Test Life Cycle, and Verification & Validation concepts,

Experiment 2

Design test cases using **Equivalence Class Partitioning** and **Boundary Value Analysis**,

Experiment 3

Test case design using **Decision Tables** and **Cause-Effect Graphing**,

Experiment 4

White Box Testing – **Statement Coverage and Branch Coverage** using sample programs,

Experiment 5

White Box Testing – **Path Testing and Cyclomatic Complexity** calculation,

Experiment 6

Unit Testing using **JUnit / PyTest** framework,

Experiment 7

Regression Testing and Test Case Prioritization techniques,

Experiment 8

Functional Testing using **Selenium WebDriver**,

Experiment 9

Test Automation – Recording and playback of test scripts,

Experiment 10

Performance Testing using **Apache JMeter** (Load, Stress, and Volume Testing),

Experiment 11

Defect Tracking and Management using **Bugzilla / JIRA (demo)**,

Experiment 12

Mini Project:

- Testing a real-time application
- Test Plan
- Test Cases
- Automation Script
- Defect Report

Tools / Software

- **Programming Languages:** Java / Python / C
- **Testing Tools:**
  - Selenium
  - JUnit / TestNG / PyTest
  - Apache JMeter
  - Bugzilla / JIRA
- **IDE:** Eclipse / IntelliJ / VS Code

  OS: Windows / Linux

Text books:

1. *Software Testing* – Naresh Chauhan, Oxford University Press
2. *Effective Software Testing* – William E, Perry, Wiley

Reference Books:

1. *Testing Computer Software* – Cem Kaner et al,, Wiley
2. *Software Testing Techniques* – Boris Beizer, Dreamtech
3. *Foundations of Software Testing* – Aditya P, Mathur, Pearson

Online Links:

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Advanced Algorithms and Optimization Techniques Laboratory

I M.Tech – II Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D25202P	PE-4	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Provide hands-on experience in implementing **advanced algorithmic techniques**,
2. Enable practical understanding of **optimization problem formulation and solution**,
3. Analyze algorithm performance and computational complexity experimentally,
4. Apply **classical and heuristic optimization techniques** to real-world problems,
5. Use modern tools and programming languages for optimization and algorithm design,

Course Outcomes: A student after completion of the course will be able to

CO1: Implement and analyze advanced algorithms using appropriate data structures,
CO2: Formulate optimization problems and solve them using numerical techniques,
CO3: Apply heuristic and metaheuristic optimization algorithms,
CO4: Evaluate algorithm efficiency through empirical analysis,
CO5: Use software tools for modeling and solving optimization problems,

List of Experiments

Experiment 1

Implementation and analysis of **Divide and Conquer algorithms** (Merge Sort / Quick Sort),

Experiment 2

Implementation of **Dynamic Programming** problems (Knapsack / Matrix Chain Multiplication),

Experiment 3

Implementation of **Greedy algorithms** (Activity Selection / Huffman Coding),

Experiment 4

Graph algorithms – **Shortest Path** (Dijkstra / Bellman-Ford),

Experiment 5

Graph algorithms – **Minimum Spanning Tree** (Prim's / Kruskal's),

Experiment 6

Implementation of **Network Flow algorithms** (Ford-Fulkerson / Edmonds-Karp),

Experiment 7

Implementation of **Linear Programming** using Simplex Method,

Experiment 8

Optimization using **Gradient Descent / Newton's Method**,

Experiment 9

Implementation of **Genetic Algorithm** for function optimization,

Experiment 10

Implementation of **Simulated Annealing / Particle Swarm Optimization**,

Experiment 11

Solving **Constraint Satisfaction Problems** using search techniques,

Experiment 12

Mini Project:

- Formulate a real-world optimization problem
- Select appropriate algorithm
- Implement and analyze results

Tools / Software

- **Programming Languages:** C / C++ / Python / Java
- **Libraries / Tools:**
  - Python (NumPy, SciPy, Matplotlib)
  - MATLAB / Octave (optional)
  - NetworkX (for graph algorithms)
- **IDE:** VS Code / PyCharm / Eclipse

  OS: Windows / Linux

Text books:

1. *Introduction to Algorithms* – Cormen et al,, MIT Press
2. *Numerical Optimization* – Nocedal & Wright, Springer

Reference Books:

1. *Convex Optimization* – Boyd & Vandenberghe
2. *Genetic Algorithms in Search, Optimization and Machine Learning* – David E, Goldberg
3. *Optimization Theory and Applications* – S, S, Rao
4. IEEE / ACM research papers on optimization techniques

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

QUANTUM TECHNOLOGIES AND APPLICATIONS

I M.Tech – II Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D58205		3	0	0	3	40	60	100

Course Objectives

1. Present core quantum principles such as superposition and entanglement without mathematical formalism.
2. Develop conceptual clarity on qubits, quantum states, and information frameworks.
3. Examine the theoretical challenges in realizing scalable quantum systems.
4. Introduce foundational ideas in quantum communication and computing.
5. Highlight applications, industrial adoption, and future research directions in quantum technologies.

Course Outcomes

Upon completion, the learner will be able to:

CO1: Explain fundamental quantum concepts conceptually.
CO2: Distinguish classical information systems from quantum information frameworks.
Identify the CO3: principal theoretical limitations in building quantum computers.
CO4: Describe the conceptual basis of quantum communication and computation.
CO5: Discuss current applications, technological trajectories, and career opportunities in the quantum domain.

UNIT-1: Foundations of Quantum Theory and Technologies
Transition from classical to quantum physics. Key conceptual principles: Superposition, Entanglement, Uncertainty, Wave-particle duality. Quantum states and measurement; the role of the observer. Representative quantum systems: electrons, photons, atoms. Concept of quantization and discrete energy levels. Strategic relevance of quantum technologies. Overview of major domains: Computing, Communication, Sensing. Global quantum initiatives: India's National Quantum Mission, EU Quantum Flagship, USA, China.
UNIT-II: Conceptual Structure of Quantum Information
Qubits: qualitative understanding using spin and polarization. Classical bits vs quantum bits: distinctions and implications. Quantum systems (non-engineering perspective): trapped ions, superconducting qubits, photonics. Coherence and decoherence mechanisms. Abstract notions: quantum states, measurement operators, Hilbert space—interpretation without mathematics. Entanglement and non-locality as foundational resources. Quantum vs classical information principles; philosophical considerations.
UNIT-III: Building a Quantum Computer – Challenges and Requirements
Conceptual prerequisites for functional quantum hardware. Fragility of quantum states: decoherence, noise, stability issues. Requirements: isolation, error resilience, scalability, control. Why maintaining entanglement is difficult; theoretical necessity of quantum error correction. Comparative overview of hardware platforms (superconducting circuits, trapped ions, photonics).Current progress vs scientific constraints; conceptual view of quantum software's role
UNIT-IV: Quantum Communication and Computing
(Redundant explanations removed, retaining only unique themes.)Quantum vs classical communication paradigms. Essentials of Quantum Key Distribution (QKD) and its security rationale. Entanglement-enabled communication protocols. Concept of the Quantum Internet and secure global networking. Introduction to quantum computing and quantum parallelism. Conceptual comparison of classical and quantum gate operations. Challenges: decoherence, noise, and the necessity of error correction frameworks.
UNIT-V: Applications, Industry, and Future Directions
Application domains: Healthcare and drug discovery, Material science and chemistry, Optimization and logistics, Quantum sensing and precision timing. Case studies: IBM, Google, Microsoft, PsiQuantum. Ethical, societal, and policy considerations. Barriers to adoption: cost, skilled workforce, standards. Emerging research and career landscapes; India's strategic opportunity in the global quantum ecosystem.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Textbooks

1. Nielsen & Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2010.
2. Rieffel & Polak, Quantum Computing: A Gentle Introduction, MIT Press, 2011.
3. Chris Bernhardt, Quantum Computing for Everyone, MIT Press, 2019.

Reference Books

1. David McMahon, Quantum Computing Explained, Wiley, 2008.
2. Kaye, Laflamme, Mosca, An Introduction to Quantum Computing, OUP, 2007. Scott Aaronson, Quantum Computing Since Democritus, CUP, 2013.
3. Susskind & Friedman, Quantum Mechanics: The Theoretical Minimum, Basic Books, 2014. Rosenblum & Kuttner, Quantum Enigma, OUP, 2011.
4. Benenti et al., Principles of Quantum Computation and Information, World Scientific, 2004. DST India and MeitY: Official Quantum Mission Reports, 2020 onwards.
5. Quantum Flagship EU: Roadmaps and Strategy Documents.

Online Learning Resources

1. IBM Quantum Experience & Qiskit Textbook Coursera – Quantum Mechanics and Quantum
2. Computation (UC Berkeley) edX – Quantum Internet & Quantum Computers YouTube – Quantum Computing for the Determined (Michael Nielsen).

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

25G3DAC201a	PEDAGOGY STUDIES	L	T	P	C
		2	0	0	0
	Semester	II			

Course Objectives:

1. To enable the students to understand the aims, rationale, policy background, and conceptual frameworks in pedagogy, curriculum, and teacher education research.
2. To develop an understanding of diverse pedagogical practices
3. To make them learn the methodologies for assessing the effectiveness of pedagogical practices and teacher education models.
4. To enable them to learn professional development strategies, including peer support, community engagement, and alignment with curriculum and assessment.

Course Outcomes(CO):Students will be able to

1. Define and explain key concepts, frameworks, and methodologies in pedagogy and teacher education research.
2. Critically analyze pedagogical practices used in diverse classroom settings, with reference to teacher education and curriculum design.
3. Evaluate the effectiveness of pedagogical approaches using quality assessment tools and theory of change models.
4. Apply evidence-based strategies to improve classroom practices, curriculum alignment, and teacher professional development.
5. Identify and address barriers to learning through innovative pedagogical strategies.
6. Design and propose research studies that contribute to filling gaps in pedagogy, curriculum, and teacher education, with focus on dissemination and impact.

UNIT -I	Foundations of Pedagogy	

Introduction to pedagogy and its importance in education - Historical and philosophical foundations of pedagogy - Theories of learning and teaching (behaviorist, cognitive, constructivist) - Role of pedagogy in shaping educational practices - Role of technology in modern pedagogy (ICT, e-learning, blended learning)

UNIT - II	Teaching-Learning Processes	

Understanding the teaching-learning process - Lesson planning and curriculum design - Strategies for effective teaching and learning (expository, collaborative, experiential) - Use of technology to enhance teaching-learning processes (multimedia, simulations, gamification)

UNIT - III	Technology Integration in Education	

Educational technology and system design - Instructional design models (ADDIE, ASSURE, Dick and Carey Model) - Emerging trends in e-learning (social learning, MOOCs, mobile learning) - ICT tools for teaching and learning (Learning Management Systems, online resources)

UNIT - IV	Pedagogy and Assessment	

Pedagogy, pedagogical analysis, and assessment - Types of assessment (placement, formative, diagnostic, summative) - Technology-based assessment tools (online quizzes, polls, discussions) - Rubrics for self and peer evaluation- Reflective Practices

UNIT - V	Contemporary Issues and Trends	

Inclusive education and technology (assistive technology, accessibility) - Change management and innovation in education - Quality assurance and evaluation in education (TQM, Six Sigma) - Future trends in pedagogy and technology (AI, AR, VR in education) - Personalized learning and adaptive teaching

Textbooks

1. Alexander, R. J. *Essays on Pedagogy*. Routledge, 2008.
2. Shulman, L. S. *The Wisdom of Practice: Essays on Teaching, Learning, and Learning to Teach*. Jossey-Bass, 2004

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Reference Books

1. *Teaching for the Future: Effective Teacher Education and Pedagogical Practices*. OECD Publishing., 2021
2. Fullan, M., & Edwards, M. *System Change in Education: Sustainability and Impact*. Routledge, 2022.
3. Coe, R., Rauch, C., Kime, S., & Singleton, D. *Great Teaching Toolkit: Evidence Review*. Evidence Based Education., 2020
4. Zeichner, K. M. *The Struggle for the Soul of Teacher Education*. Routledge, 2024
5. UNESCO. *Global Education Monitoring Report: Pedagogy, Teachers and Learning*. UNESCO Publishing, 2024
6. Hattie, J. *Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement*. Routledge., 2009
7. Darling-Hammond, L. *Teacher Education Around the World: What Can We Learn from International Practice?* Routledge, 2007

Online Resources

- **UNESCO Education Resources** – https://www.unesco.org/education
- **OECD Education and Skills** – https://www.oecd.org/education
- **ERIC (Education Resources Information Center)** – https://eric.ed.gov (peer-reviewed papers, reports).
- **World Bank Education** – https://www.worldbank.org/en/topic/education (research reports on teacher development in developing countries).
- **NPTEL/SWAYAM MOOCs** – Teacher education and pedagogy-focused courses.
- **Google Scholar Alerts** – set alerts for "pedagogical practices", "teacher education", "curriculum research" for the latest academic papers.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

25G3DAC201b	PERSONALITY DEVELOPMENT THROUGH LIFE ENLIGHTENMENT SKILLS	L	T	P	C
		2	0	0	0
	Semester	**II**			

CourseObjectives:This course will enable students:

- To develop students' self-awareness by identifying their strengths, weaknesses, opportunities, and challenges (SWOC analysis).
- To enable students to understand and apply the principles of emotional intelligence and effective interpersonal communication.
- To cultivate positive thinking, resilience, mindfulness, and a growth-oriented mindset.
- To enhance verbal and non-verbal communication skills, including confidence in public speaking and professional presentations.
- To familiarize students with leadership styles, teamwork strategies, and collaborative problem-solving in personal and professional contexts.

Course Outcomes(CO):Student will be able to

1. Define and explain key concepts of self-awareness, personality, and personal growth.
2. Identify and apply strategies of emotional intelligence to regulate emotions and build effective interpersonal relationships
3. Demonstrate positive thinking, gratitude, and resilience to overcome personal and professional challenges
4. Analyze barriers to effective communication and apply verbal and non-verbal communication techniques in diverse contexts.
5. Prepare, deliver, and evaluate effective presentations and public speeches with confidence
6. Develop leadership and teamwork skills to collaborate, negotiate, and solve problems in group settings.

UNIT –I	**Self-Awareness and Personal Growth**	

Understanding personality and its development- Identifying strengths, weaknesses, opportunities, and challenges (SWOC analysis)- Setting personal and professional goals- Practicing Self-Reflection and Journaling
(Activities: Personalitya assessments, self reflection exercises, group discussions, SWOC analysis worksheet, Action Plan, SMART goal activities, Reflective journaling, Self-care Planning)

UNIT – II	**Emotional Intelligence and Interpersonal Skills**	

Understanding emotional intelligence and its importance - Developing self-awareness, self-regulation, and motivation - Building effective communication and interpersonal skills
- Conflict resolution and negotiation strategies.
(Activities: Emotional Intelligence Quiz, Self-Reflection exercises, feedback sessions, mindfulness exercises, Positive self-talk, Active Listening exercises, conflict-resolution Role-play, Case studies & Group activities)

UNIT – III	**Positive Thinking and Attitude**	

Understanding the power of positive thinking- Developing a growth mindset and resilience - Practicing gratitude and mindfulness- Overcoming negative thoughts and behaviors
(Activities on positive thinking, growth mindset, mindfulness and self-care plan for overcoming negative thoughts)

UNIT – IV	**Effective Communication and Presentation Skills**	

Understanding the importance of effective communication- Developing verbal and non-verbal communication skills- Preparing and delivering effective presentations- Building confidence and public speaking skills
(Activities: Group discussions, Case studies, Role-Play, Non-verbal communication exercises, Practice presentations, Peer feedback, Public speaking exercises, Storytelling, Debates)

UNIT – V	**Leadership and Teamwork**	

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Understanding leadership styles and qualities - Developing leadership skills and qualities- Building effective teams and teamwork strategies- Practicing collaboration and problem-solving
(Activities: Case studies, Group discussions, Debates, Leadership role-playing, team building activities, Group projects, Collaborative problem-solving exercies, feedback sessions)

Prescribed Books

1. Daniel Goleman,Emotional Intelligence: Why It Can Matter More Than IQ, Bantam Books, 2017.
2. Stephen R. Covey, The 7 Habits of Highly Effective People, Simon & Schuster, 2020.

Reference Books

1. Dale Carnegie, How to Win Friends and Influence People, Simon & Schuster, 2020.
2. Brian Tracy, Goals!: How to Get Everything You Want Faster Than You Ever Thought Possible, Berrett-Koehler Publishers, 2021.
3. Robin Sharma, The 5 AM Club: Own Your Morning, Elevate Your Life, HarperCollins, 2020.
4. Carol S. Dweck, Mindset: The New Psychology of Success, Random House, 2016.
5. Daniel H. Pink,Drive: The Surprising Truth About What Motivates Us, Riverhead Books, 2018.
6. John C. Maxwell, Leadershift: 11 Essential Changes Every Leader Must Embrace, HarperCollins, 2019.

Online Resources

1. Coursera – Personal Development Specialization (https://www.coursera.org)
2. edX – Leadership and Emotional Intelligence Courses (https://www.edx.org)
3. FutureLearn – Mindfulness and Resilience Training (https://www.futurelearn.com)
4. MindTools – Practical resources on leadership, communication, and emotional intelligence (https://www.mindtools.com)
5. Positive Psychology – Articles and tools on resilience, gratitude, and well-being (https://positivepsychology.com)
6. TED Talks – Inspirational talks on leadership, communication, and self-growth (https://www.ted.com)
7. Harvard Business Review (HBR) – Leadership, negotiation, and workplace communication (https://hbr.org)

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

25G3DAC201c	YOGA FOR STRESS MANAGEMENT	L	T	P	C
		2	0	0	0
	Semester	II			

Course Objectives:

1. To make the students understand the foundational concepts of Yoga, including Ashtanga (eight limbs) as prescribed in classical texts.
2. To enable them analyze the principles of Yama and Niyama, and their role in ethical, personal, and social development.
3. To make them learn do's and don'ts of life through the practice of ahimsa, satya, astheya, brahmacharya, aparigraha, shaucha, santosh, tapa, swadhyaya, and ishwar-pranidhana.
4. To make them practice asanas and pranayama techniques for physical fitness, mental balance, and spiritual awareness.
5. To make them understand the holistic lifestyle through regular yoga practice, leading to personality development.

Course Outcomes(CO):Student will be able to

1. Explain the eight limbs of Yoga (Ashtanga) and their interrelationship in holistic development.
2. Demonstrate understanding of Yama and Niyama as ethical guidelines and apply them in personal and professional life.
3. Differentiate between do's and don'ts in daily life with reference to Yogic principles like ahimsa, satya, and swadhyaya.
4. Perform selected asanas and pranayama techniques with correct posture, breathing, and awareness.
5. Evaluate the physical, mental, and emotional benefits of yoga practices in stress reduction, concentration, and self-discipline.
6. Integrate yoga philosophy and practices into a personal routine for sustainable health and inner growth.

UNIT -I		
Definitions of Eight parts of yoga.(Ashtanga)		
UNIT - II		
Yam and Niyam.		
UNIT - III		
Do's and Don'tsin life.		
i) Ahinsa,satya,astheya,bramhacharya and aparigraha ii)Shaucha,santosh,tapa,swadhyay,ishwarpranidhan		
UNIT - IV		
Asanand Pranayam		
UNIT - V		
i) Various yoga poses and their benefits for mind and body ii) Regularization of breathing techniques and its effects-Types of pranayam		

Textbooks

1. Swami Prabhavananda and Christopher Isherwood (translation & commentary), *Patanjali Yoga Sutras*,Sri Ramakrishna Math, 1953.
2. B.K.S. Iyengar, *Light on Yoga*, Thorsons, 2003.

Reference Books

1. T.K.V. Desikachar, *The Heart of Yoga: Developing a Personal Practice*, Inner Traditions 2nd Edition, 1999.

2. Acharya Yatendra, *Yoga & Stress Management*, Fingerprint Publishers, 2019

3. Yamini Muthanna, *The Power of Yoga*, Om Books International, 2015.

4. Nayaswami Devarshi, *Kriya Yoga: Spiritual Awakening for the New Age*, Ananda Sangha Publications, 2023.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Online Resources

- NPTEL / SWAYAM Online Courses – Yoga and Physical Education modules.
- AYUSH Ministry Website: https://yoga.ayush.gov.in – official yoga resources, protocols, and research.
- Yoga Journal: https://www.yogajournal.com – practical guides, research updates, asana tutorials.
- Art of Living Foundation: https://www.artofliving.org – pranayama, meditation, and wellness practices.
- YouTube Channels (scholarly & practice-based):
    - Sivananda Yoga Vedanta Centre
    - Yoga with Adriene (for practical asana guidance)

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SERVICE ORIENTED ARCHITECTURE

I M.Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D25301a	PE-5	3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Understand SOA and evolution of SOA.
2. Understand web services and primitive, contemporary SOA.
3. Understand various service layers.
4. Understand service-oriented analysis and design based on guidelines.

Course Outcomes: A student after completion of the course will be able to

CO1: Comprehend the need for SOA and its systematic evolution
CO2: Apply SOA technologies to enterprise domain
CO3: Design and analyse various SOA patterns and techniques
CO4: Compare and evaluate best strategies and practices of SOA

UNIT-1
Introducing SOA: Fundamental SOA, Common Characteristics of Contemporary SOA, Common Tangible Benefits of SOA, Common Pitfalls of Adopting SOA. The Evolution of SOA: An SOA Timeline, The Continuing Evolution of SOA, The Roots of SOA.
UNIT-1I
Web Services and Primitive SOA: The Web Services Frame Work, Services, Service Descriptions, Messaging. Web Services and Contemporary SOA (Part I-Activity management and Composition): Message Exchange Patterns, Service Activity, Coordination, Atomic Transactions, Orchestration, and Choreography. Web Services and Contemporary SOA (Part-II-Advanced Messaging, Metadata and Security): Addressing, Reliable Messaging, Correlation, Policies, Metadata exchange, Security.
UNIT-III
Principles of Service-Orientation: Service–Orientation and the Enterprise, Anatomy of SOA, Common Principles of Service–Orientation, Interrelation between Principles of Service- Orientation, Service Orientation and Object Orientation, Native Web Services Support for Principles of Service-Orientation. Service Layers: Service-Orientation and Contemporary SOA, Service Layer abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration Scenarios.
UNIT-IV
SOA Delivery Strategies: SOA Delivery Lifecycle Phases, The Top-Down Strategy, The Bottom-up Strategy, The Agile Strategy. Service Oriented Analysis (Part I-Introduction): Introduction to Service Oriented Analysis, Benefits of a Business Centric SOA, Deriving Business Services. Service Oriented Analysis (Part-II-Service Modelling): Service Modelling, Service Modelling Guidelines, Classifying Service Model Logic, Contrasting Service Modelling Approaches. Service Oriented Design (Part I-Introduction): Introduction to Service-Oriented Design, WSDL Related XML Schema Language Basics, WSDL Language Basics, Service Interface Design Tools. Service Oriented Design (Part II-SOA Composition Guidelines): SOA Composing Steps, Considerations for Choosing Service Layers, Considerations for Positioning Core SOA Standards, Considerations for Choosing SOA Extensions.
UNIT-V
Service Oriented Design (Part III- Service Design): Service Design Overview, Entity- Centric Business Service Design, Application Service Design, Task-Centric Business Service Design, Service Design Guidelines. Service Oriented Design (Part IV-Business Process Design): WS-BPEL Language Basics, WS- Coordination Overview, Service Oriented Business Process Design.

Text books:

1. Service-Oriented Architecture-Concepts, Technology, and Design, Thomas Erl, Pearson Education, 2006.
2. Understanding SOA with Web Services, Eric Newcomer, Greg Lomow, Pearson Education, 2005.

Reference Books:

1. Thomas Erl; Service Oriented Architecture Concepts Technology & Design, Pearson Education Limited; 2015, ISBN-13: 9788131714904.
2. 2 Guido Schmutz, Peter Welkenbach, Daniel Liebhart; Service Oriented Architecture An Integration Blueprint; Shroff Publishers & Distributors; 2010, ISBN-13: 9789350231081

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Cyber-Physical Systems

I M.Tech – III Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D25301b	PE-5	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Introduce the concepts and foundations of **Cyber-Physical Systems**.
2. Explain the integration of **computation, communication, and physical processes**.
3. Provide understanding of **modeling, design, and analysis** of CPS.
4. Discuss **real-time, safety, and security issues** in CPS.
5. Explore **applications of CPS** in modern engineering domains.

Course Outcomes: A student after completion of the course will be able to

CO1: Understand architecture and components of cyber-physical systems.
CO2: Model and analyze CPS using appropriate techniques.
CO3: Design CPS with real-time and control constraints.
CO4: Identify safety, security, and reliability challenges in CPS.
CO5: Apply CPS concepts to real-world application domains

UNIT-1
Introduction to Cyber-Physical Systems: Definition and Characteristics of CPS,CPS vs Embedded Systems, CPS Architecture, Sensors and Actuators, Physical Processes and Dynamics, Cyber Components, Feedback Control Basics, CPS Design Challenges, Case Studies of CPS.
UNIT-1I
Modeling and Design of CPS: Modeling of Physical Systems, Continuous and Discrete Models, Hybrid Systems, State-Space Modeling, Modeling Tools (MATLAB/Simulink, Modelica), Co-Design of Hardware and Software, Time and Event-Driven Models, CPS Design Methodologies, Verification and Validation Basics.
UNIT-III
Communication and Real-Time Aspects: CPS Networking Requirements, Wired and Wireless Communication, Real-Time Operating Systems (RTOS), Scheduling Algorithms, Timing Constraints, Clock Synchronization, Middleware for CPS, Distributed CPS, Quality of Service (QoS).
UNIT-IV
Safety, Security, and Reliability: Safety-Critical CPS, Fault Models and Fault Tolerance, Reliability Analysis, CPS Security Threats, Secure Communication, Intrusion Detection in CPS, Privacy Issues, Risk Assessment, Standards for CPS Safety and Security.
UNIT-V
CPS Applications and Emerging Trends: Smart Grids, Smart Transportation Systems, Industrial Automation (Industry 4.0), Healthcare CPS, Smart Cities, Autonomous Systems, Internet of Things (IoT) and CPS, AI in CPS, Future Research Directions.

Text books:

1. **Edward A. Lee & Sanjit A. Seshia**, *Introduction to Embedded Systems – A Cyber-Physical Systems Approach*, MIT Press
2. **Rajeev Alur**, *Principles of Cyber-Physical Systems*, MIT Press

Reference Books:

1. **Herman Kopetz**, *Real-Time Systems: Design Principles for Distributed Embedded Applications*
2. **Karl-Erik Årzén et al.**, *Modeling and Design of Cyber-Physical Systems*
3. **Jan Lunze**, *Control Theory of Digitally Networked Dynamic Systems*

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

AI-Enabled Software Quality Assurance

I M.Tech – III Semester								SJCET-R25
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
25G3D25301c	PE-5	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	0	0	3	40	60	100

Course Objectives: This course is designed to enable the students

1. Introduce **AI and Machine Learning concepts** applied to software quality assurance.
2. Explain how **intelligent techniques enhance software testing and verification**.
3. Provide knowledge of **AI-driven defect prediction and test automation**.
4. Enable students to analyze **software quality metrics using data-driven approaches**.

Course Outcomes: A student after completion of the course will be able to

CO1: Understand the role of AI in modern software quality assurance.
CO2: Apply machine learning techniques for defect prediction and fault localization.
CO3: Design AI-based test automation frameworks.
CO4: Evaluate software quality using predictive analytics and intelligent metrics.
CO5: Analyse challenges, ethics, and future directions of AI-enabled SQA.

UNIT-1
Foundations of Software Quality Assurance: Software Quality Concepts and Definitions, Software Quality Models (ISO, McCall, Boehm), Software Testing Life Cycle (STLC), Verification and Validation, Manual vs Automated Testing, Limitations of Traditional SQA, Introduction to AI in Software Engineering, Motivation for AI-EnabledSQA, Case Studies in SQA
UNIT-1I
Machine Learning for Software Quality: Data Collection from Software Repositories, Software Metrics (Code, Process, Product), Supervised and Unsupervised Learning, Defect Prediction Models, Classification Algorithms (Decision Trees, SVM, Random Forest), Feature Selection Techniques, Model Evaluation Metrics, Handling Imbalanced Data, Tools for ML-based SQA.
UNIT-III
AI-Driven Software Testing: Intelligent Test Case Generation, Search-Based Software Testing, Genetic Algorithms for Testing, Reinforcement Learning in Test Optimization, AI-Based Test Prioritization, Regression Test Selection, Test Oracle Problem, Autonomous Testing Systems, AI Tools for Test Automation.
UNIT-IV
Defect Analysis and Quality Prediction: Defect Classification and Root Cause Analysis, Fault Localization Techniques, Predictive Quality Analytics, Deep Learning for Bug Detection, Natural Language Processing for Bug Reports, Sentiment Analysis in Issue Tracking, Anomaly Detection in Software Systems, Continuous Quality Monitoring, Industrial Case Studies.
UNIT-V
Advanced Topics and Emerging Trends: AI in DevOps and Continuous Testing, Self-Healing Systems, AI-Based Code Review and Refactoring, Explainable AI for SQA, Ethics and Bias in AI Models, Security Testing with AI, Quality Assurance for AI Systems, Tool Chains and Frameworks, Future Research Directions.

Text books:

1. **Rana, Rajasekaran**, *AI for Software Quality Engineering*, Springer
2. **Jinhan Kim et al.**, *Machine Learning for Software Engineering*, Springer

Reference Books:

1. **Tom M. Mitchell**, *Machine Learning*, McGraw-Hill
2. **Ian Sommerville**, *Software Engineering*, Pearson
3. **Boris Beizer**, *Software Testing Techniques*
4. IEEE / ACM Transactions on Software Engineering

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GREEN BUILDINGS

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D20302	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

The objectives of this course are:

1. To understand the fundamental concepts of green buildings, their necessity, and sustainable features.
2. To analyze green building concepts, rating systems, and their benefits in India. (Analyzing - Level 4)
3. To apply green building design principles, energy efficiency measures, and renewable energy sources.
4. To evaluate air conditioning systems, HVAC designs, and energy modeling for sustainable buildings.
5. To assess material conservation strategies, waste management, and indoor environmental quality in green buildings

Course Outcomes (COs)

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

CO1: Understand the importance of green buildings, their necessity, and sustainable features.

CO2: Analyze various green building practices, rating systems, and their impact on environmental Sustainability.

CO3: Apply principles of green building design to enhance energy efficiency and incorporate renewable energy Sources.

CO4: Evaluate HVAC systems, energy-efficient air conditioning techniques, and their role in sustainable building design.

CO5:Assess material conservation techniques, waste reduction strategies, and indoor air quality management in green buildings.

UNIT-1
Introduction to Green Building – Necessity of Green Buildings, Benefits of Green Buildings, Green Building Materials and Equipment in India, Key Requisites for Constructing a Green Building, Important Sustainable Features for Green Buildings.
UNIT-II
Green Building Concepts and Practices – Indian Green Building Council, Green Building Movement in India, Benefits Experienced in Green Buildings, Launch of Green Building Rating Systems, Residential Sector, Market Transformation; Green Building Opportunities and Benefits: Opportunities of Green Buildings, Green Building Features, Material and Resources, Water Efficiency, Optimum Energy Efficiency, Typical Energy-Saving Approaches in Buildings, LEED India Rating System, and Energy Efficiency.
UNIT-III
Green Building Design – Introduction, Reduction in Energy Demand, Onsite Sources and Sinks, Maximizing System Efficiency, Steps to Reduce Energy Demand and Use Onsite Sources and Sinks, Use of Renewable Energy Sources, Eco-friendly Captive Power Generation for Factories, Building Requirements.
UNIT-IV
Air Conditioning – Introduction, CII Godrej Green Business Centre, Design Philosophy, Design Interventions, Energy Modeling, HVAC System Design, Chiller Selection, Pump Selection, Selection of Cooling Towers, Selection of Air Handling Units, Pre-cooling of Fresh Air, Interior Lighting Systems, Key Features of the Building, Eco-friendly Captive Power Generation for Factories, Building Requirements.
UNIT-V
Material Conservation – Handling of Non-Process Waste, Waste Reduction During Construction, Materials with Recycled Content, Local Materials, Material Reuse, Certified Wood, Rapidly Renewable Building Materials and Furniture. Indoor Environment Quality and Occupational Health – Air Conditioning, Indoor Air Quality, Sick Building Syndrome, Tobacco Smoke.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Text Books:

1. Handbook on Green Practices published by Indian Society of Heating Refrigerating and Air conditioning Engineers, 2009.
2. Green Building Hand Book by Tomwoolley and Samkimings, 2009.

Reference :-

1. Complete Guide to Green Buildings by Trish riley
2. Standard for the design for High Performance Green Buildings by Kent Peterson, 2009
3. Energy Conservation Building Code –ECBC-2020, published by BEE

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ROAD SAFETY ENGINEERING

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D93203b	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

1. Discusses the fundamentals causes of road accidents & some of the statistical methods To analyze the traffic safety.
2. The accident investigation and risk management are dealt.
3. Introduction of Road Safety as an integral part of Road Design is emphasized.

Course Outcomes (COs)

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

CO1: To understand the basic causes of road accidents
CO2: To conduct accident investigations and identify black spots
CO3: To design & plan road geometrics incorporating road safety
CO4: To understand the importance of Road Signs and Markings
CO5: To conduct Road Safety Audit

UNIT-1
Accident Investigations and Risk Management, Collection of Accident Data, Assessment of Road Safety, Methods to Identify and Prioritize Hazardous Locations and Elements, Determine Possible Causes of Crashes, Crash Reduction Capabilities and Countermeasures, Effectiveness of Safety Design Features, Accident Reconstruction, Condition and Collision Diagram.
UNIT-II
Traffic Engineering Studies; Statistical Methods In Traffic Safety Analysis –Regression Methods, Poisson Distribution, Chi- Squared Distribution, Statistical Comparisons- Traffic Management Measures And Their Influence On Accident Prevention.
UNIT-III
Road Safety in ` Transport Planning and Geometric Design: Vehicle and Human Characteristics, Road Design and Safety Elements, Redesigning Junctions, Cross Section Improvements, Traffic Control, Traffic Calming Measures, Road Safety Furniture
UNIT-IV
Role of Signs and Markings in ` Safety: Types of Signs – Design Specifications – Guidelines for Installation – Role of Signs in `Safety; Types of Road Markings – Design Specifications – Role of Road Markings in `Safety.
UNIT-V
Traffic Management Systems for Safety, Road Safety Audits and Tools for Safety Management Systems, Road Safety Audit Process, Road Safety Improvement Strategies, ITS and Safety.

Textbooks:

1. Traffic Engineering and Transportation Planning – L.R. Kadiyali, Khanna Publishers
2. Fundamentals of Transportation Engineering - C.S.Papacostas, Prentice Hall India.
3. Road Safety by NCHRP

Reference Books:

1. Transportation Engineering – An Introduction, C.Jotin Khisty, B. Kent Lall
2. Fundamentals of Traffic Engineering, Richardo G Sigua
3. Handbook of Road Safety Measures, Second Edition, Rune Elvik, Alena Hoye, TrulsVaa, Michael Sorenson

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

IOT AND ITS APPLICATIONS

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D06302	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

To apply the Knowledge in IOT Technologies and Data management.

1. To determine the values chains Perspective of M2M to IOT.
2. To implement the state of the Architecture of an IOT.
3. To compare IOT Applications in Industrial & real world.
4. To demonstrate knowledge and understand the security and ethical issues of an IOT.

Course Outcomes (COs)

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

CO1: Apply the Knowledge in IOT Technologies and Data management.
CO2: Determine the values chains Perspective of M2M to IOT.
CO3: Implement the state of the Architecture of an IOT.
CO4: Compare IOT Applications in Industrial & real world.
CO5: Demonstrate knowledge and understand the security and ethical issues of an IOT.

UNIT-1
Fundamentals of IoT: Evolution of Internet of Things, Enabling Technologies, IoT Architectures, oneM2M, IoT World Forum (IoTWF) and Alternative IoT models, Simplified IoT Architecture and Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects. IoT Platform overview: Overview of IoT supported Hardware platforms such as: Raspberry pi, ARM Cortex Processors, Arduino and Intel Galileo boards.
UNIT-II
IoT Protocols: IT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and Lora WAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks, Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks, Application Transport Methods: Supervisory Control and Data Acquisition, Application Layer Protocols: CoAP and MQTT.
UNIT-III
Design and Development: Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks, Arduino, Board details, IDE programming, Raspberry Pi, Interfaces and Raspberry Pi with Python Programming.
UNIT-IV
Data Analytics and Supporting Services: Structured Vs Unstructured Data and Data in Motion Vs Data in Rest, Role of Machine Learning – No SQL Databases, Hadoop Ecosystem, Apache Kafka, Apache Spark, Edge Streaming Analytics and Network Analytics, Xively Cloud for IoT, Python Web Application Framework, Django, AWS for IoT, System Management with NETCONF-YANG.
UNIT-V
Case Studies/Industrial Applications: IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments. Use of Big Data and Visualization in IoT, Industry 4.0 concepts. Sensors and sensor Node and interfacing using any Embedded target boards (Raspberry Pi / Intel Galileo/ARM Cortex/ Arduino).

Textbooks:

1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017.
2. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press,2015

Reference Books:

1. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012 (for Unit 2).
2. ―From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence‖, Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.
3. Architecting the Internet of Things, Dieter Uckelmann, Mark Harrison, Michahelles and Florian (Eds), Springer, 2011.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PHOTOVOLTAIC SYSTEMS

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
25G3D07302	OE-1	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100

Course Objectives (COs)

1. To introduce photovoltaic systems.
2. To deal with various technologies of solar PV cells.
3. To understand details about manufacture, sizing and operating techniques.
4. To have knowledge of design considerations.

Course Outcomes (COs)

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

CO1: Identify photovoltaic system components and system types.

CO2: Calculate electrical energy and power.

CO3: Correctly size system components, design considerations of solar equipment.

CO4: Design a basic grid-tie PV system.

UNIT-1: SOLAR ENERGY
Sun and Earth, Solar Spectrum, Solar Geometry, Solar radiation on horizontal and inclined planes, Instruments for measurement of solar radiation, Solar cell, Equivalent circuit, V-I characteristics, Performance improvement.
UNIT-II: SOLAR CELLS
Manufacture of Solar Cells-Technologies, Design of Solar cells, Photovoltaic modules, Design requirements, Encapsulation systems, Manufacture, Power rating, Hotspot effect, Design qualifications.
UNIT-III: PROTECTION AND MEASUREMENTS
Flat plate arrays, Support structures, Module interconnection and cabling, Lightning protection, Performance measurement using natural sun light and simulator, Determination of temperature coefficients, Internal series resistance, Curve correction factor.
UNIT-IV: PHOTOVOLTAIC SYSTEMS
Photovoltaic systems, Types, General design considerations, System sizing, Battery sizing, Inverter sizing, Design examples, Balance of PV systems.
UNIT-V: MAXIMUM POWER POINT TRACKERS
Maximum power point trackers, Perturb and observe, Incremental conductance method, Hill climbing method, Hybrid and complex methods, Data based and other approximate methods, Instrument design, Other MPP techniques, Grid interactive PV system.

Textbooks:

1. F.C.Treble, —Generating electricity from Sun‖, Pergamon Press.
2. A.K.Mukherjee, Nivedita Thakur,‖ Photovoltaic systems: Analysis and design‖, PHI, 2011.

Reference Books:

1. C.S.Solanki,‖ Solar Photovoltaic's: Fundamentals, Technologies and applications‖, PHI, 2009.

Online Learning Resources:

1. https://nptel.ac.in/courses/117108141

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INTEGRATED PRODUCT DESIGN AND DEVELOMENT

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D15302	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

1. To know the design concepts, product development planning and customer requirements
2. To generate the concept using creative problem solving methods, concept generation and testing
3. To understand product teardown, specifications, portfolios and Architecture, configurations and detailed design with prototyping
4. To acquaint DFN process and know design aspects for risk, reliability and safety and environments
5. To do industrial design, intellectual property and robust design and product development economics

Course Outcomes (COs)

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

CO1: Understand To know the design concepts, product development planning and customer requirements

CO2: Generate the concept using creative problem solving methods, concept generation and testing

CO3: Realize Product Teardown, Specifications, Portfolios and Architecture, configurations and prototyping

CO4: Demonstrate and classify The Design aspects for Risk, Reliability and Safety and environments

CO5: Do industrial design, understand intellectual property, robust design and product development economics

UNIT-1: PRODUCT DEVELOPMENT CONCEPTS
Design Concepts: Design process – Considerations of a good design – Description of good design process – Design codes and standards. Product Development and Planning Process: Characteristics and challenges of product development – Concept development – Generic product development – Product development process flows –Tyco product development – Product development organizations – Organizational structure and design – Product and process cycles – Technological innovation – Structure of opportunity – Opportunity identification – Product planning process – Types of product development projects. Identifying Customer Needs: Process of identifying customer needs – Customer requirements.
UNIT-II: CONCEPT SELECTION, GENERATION AND TESTING
Concept Generation: Activity – Concept generation process – Creativity and problem solving – Creative thinking methods and design – Functional decomposition and synthesis – Morphological methods – Axiomatic design. Concept Selection and Testing: Development process – Choosing a concept – Concept screening and scoring – Decision making and evaluation – Methods for testing product concepts.
UNIT-III: EMBODIMENT AND DETAIL DESIGN
Product Teardown, Specifications, Portfolios and Architecture: Teardown process, methods and applications – Post teardown report – Benchmarking approach and support tools for benchmarking process – Product portfolios architecture – Architecture type – Platform architecture – Target Specifications – Setting the final specifications – Modularity – Implications of the architecture – Establishing the architecture – Delayed differentiation – Platform planning – Related system-level design. Configuration and Detail Design: Generating, analyzing and evaluating configuration design – Best practices for configuration design – Design for X – Design and manufacturing information – Final design review – Activities beyond detail design. Prototyping: Principles, types and technologies – Understanding prototypes and planning.
UNIT-IV: DESIGN FOR ENVIRONMENT, MANUFACTURING AND SAFETY
Design for Manufacture and Environment: Cross-functional team – Overview of DFM process – Life cycles – Environmental impacts – Design for environment process. Design for Risk, Reliability and Safety: Classification of societal hazards – Standards – Risk assessment – Design for reliability – Causes of unreliability – Minimizing failure – FMEA – Fault tree analysis – Defects and failure modes – Potential dangers – Guidelines for design for safety – Warning labels.
UNIT-V: INDUSTRIAL DESIGN AND PRODUCT DEVELOPMENT ECONOMICS
Industrial Design: Need – Impact – Industrial design process – Management of the industrial design process – Assessing the quality of industrial design. Robust Design: Robust design process. Intellectual Property: Disclosure – Process of pursuing a patent. Product Development Economics: Elements of economic analysis and

process. Managing Projects: Understanding and representing tasks – Baseline project planning – Accelerating projects – Project execution – Postmortem project evaluation - Project Portfolio Management (PPM) - Earned Value Management (EVM) - Sustainability and ESG in Project Management.

Textbooks:

1. Product Design and Development, Karl T Ulrich, Steven D Eppinger and Maria C. Yang, 7/e, 2020,McGraw-Hill Education Pvt.Ltd., Noida.
2. Engineering Design, George E.Dieter and Linda C.Schmidt, 4/e, 2013, McGraw-Hill Education Pvt.Ltd.,Noida.

Reference Books:

1. Product Design, Kevin Otto and Kristin Wood, 1/e, 2003, Pearson Education, India.
2. Product Development, Anil Mital, Anoop Desai, Anand Subramanian and Aashi Mital, 1/e, 2007, Butterworth-Heinemann, Elsevier.
3. Integrated Product and Process Design and Development: the Product Realization Process (Special Indian Edition), Edward B. Magrab, Satyandra K. Gupta, F. Patrick McCluskey and Peter A. Sandborn, 2/e, 2010, CRC Press, Taylor & Francis Group, LLC.
4. Product Design for Engineers, Devdas Shetty, 1/e, 2016, Cengage Learning, India.
5. Introduction to Product Design and Development for Engineers, Ali Jamnia, 2018, CRC Press,Taylor & Francis Group, LLC.

Online Learning Resources:

1. https://onlinecourses.nptel.ac.in/noc25_me121/preview
2. https://onlinecourses.swayam2.ac.in/imb25_mg123/preview

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ADVANCED NUMERICAL METHODS AND COMPUTATIONAL MATHEMATICS

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
25G3DOE301a	OE-1	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100

Course Objectives (COs)

1. Equip students to analyze, design, and implement stable, accurate, and efficient numerical algorithms for large- scale models across engineering domains.
2. Develop mastery in discretization, linear algebra, and optimization techniques, supported by modern
3. scientific computing tools and reproducible workflows.
4. Integrate high- performance computing paradigms to solve realistic, data- and compute- intensive problems.

Course Outcomes (COs)

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

CO1: Analyze conditioning, stability, and error behavior of numerical linear algebra routines for dense and sparse
Problems (L3, L4).

CO2: Derive and apply finite difference for BVP for linear and nonlinear models (L3, L4, L5).

CO3: Derive and apply finite difference for PDE models; assess consistency and stability with Lax equivalence and CFL ideas(L3, L4).

CO4: Optimize engineering models via unconstrained/constrained methods; justify algorithm selection and tuning using theory and diagnostics (L4, L5).

CO5: Integrate GPU/MPI paradigms and the SciPy stack to build scalable, reproducible computational workflows; profile and validate results (L4, L5).

UNIT-1: Numerical Linear Algebra
Floating- point arithmetic, conditioning, stability, backward/forward error. • Direct solvers: LU/Cholesky/QR; pivoting and orthogonality. • SVD, low- rank approximations, pseudoinverses. • Iterative solvers: Jacobi/Gauss–Seidel/CG/GMRES; basic preconditioning. • Sparse matrix formats and operations for large- scale systems. • Krylov subspace and Arnoldi/Lanczos overviews.
UNIT-II: Discretization of linear and nonlinear ODEs
Numerical solutions of initial value problems: (single step and multi -step methods) • Stability of the numerical methods for initial value problems. • Boundary value problems; shooting method. • Finite difference for BVP (second and higher order methods), for linear and nonlinear problems
UNIT-III: Discretization of linear and nonlinear PDEs
Finite difference methods for parabolic problems explicit and implicit methods, • Finite difference for hyperbolic explicit and implicit methods, Elliptic PDE. R25 Regulations • Consistency, Stability and convergence. • Lax equivalence and CFL (Courant–Friedrichs–Lewy) ideas. • Iterative solvers and multigrid overview for discretized systems.
UNIT-IV: Numerical Optimization and Inverse Problems
Unconstrained methods: gradient, Newton, quasi- Newton, trust- region. • Constrained methods: KKT conditions, interior- point, SQP principles. • Nonlinear least squares: Gauss–Newton and Levenberg–Marquardt. • Regularization (Tikhonov/L1) and model selection concepts. • Scientific Python stack:scipy.optimize.minimize, scipy.optimize.linprog, and scipy.optimize.curve_fit; solver choice and diagnostics.
UNIT-V: High- Performance Scientific Computing
Parallel paradigms: data/task parallelism; domain decomposition fundamentals. • GPU programming model: threads/warps/memory hierarchy; CUDA libraries. • MPI and PETSc for scalable sparse linear algebra and time- steppers. • Performance engineering: profiling, locality, and roofline- style thinking. • Python at scale: NumPy/SciPy sparse andvectorization; brief Numba/CuPy ecosystem. • End- to- end case sketches in CFD/structures/machine learning numerics.

Textbooks:

1. Trefethen, L. N., and D. Bau III, Numerical Linear Algebra, Twenty- Fifth Anniversary Edition, SIAM, 2023.
2. LeVeque, R. J., Finite Difference Methods for Ordinary and Partial Differential Equations: Steady- State and Time- Dependent Problems, SIAM, 2007.
3. Nocedal, J., and S. J. Wright, Numerical Optimization, 2nd ed., Springer, 2006.

References:

1. Hwu, W.- M. W., and D. B. Kirk, Programming Massively Parallel Processors: A Hands- on Approach, 4th ed., Elsevier, 2022.
2. B. N. dutta, ―Numerical linera algebra and Application, - Springer Publications
3. SciPy Project. ―Optimization (scipy.optimize) — SciPy Manual‖ and ―Optimization and Root Finding — SciPy 1.16.2 (stable) Reference.‖
4. Balay, S., et al. PETSc/TAO Users Manual. Argonne National Laboratory, ANL-21/39 Rev 3.18release documentation.
5. Logg, A., K.- A. Mardal, and G. N. Wells (eds.). Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book. Berlin: Springer, 2012.
6. David Kincaid Ward Chenery, ― Numerical analysis Mathematics of scientific computing, - AMS Book publishers.
7. M. K. Jain, S. R. K. Iyengar and R. K. Jain, ― Computational Methods for Partial Differential Equations, - New Age publications.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MATHEMATICS FOR MACHINE LEARNING AND DATA SCIENCE

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3DOE301b	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

1. Equip students with rigorous linear algebra, optimization, and probabilistic tools that underpin modern ML and data science across engineering domains.
2. Enable cross- disciplinary modeling, inference, and decision- making using mathematically principled methods and uncertainty quantification.
3. Develop capability to analyze, formulate, and validate algorithms with provable properties and computational efficiency for real- world applications.

Course Outcomes (COs)

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

CO1: Analyze linear models and matrix factorizations for dimensionality reduction and feature Representation (L3, L4)

CO2: Formulate and solve convex and constrained optimization problems; compare first- and Second- order methods (L4,L5)

CO3: Apply probabilistic modeling, Bayesian inference, and information- theoretic measures to estimation and generalization (L3, L4, L5)

CO4: Prove convergence properties or error bounds for selected learning algorithms such as SGD and regularized estimators (L4, L5)

CO5: Evaluate and integrate methods into applications such as PCA, SVMs, and neural networks with appropriate regularization (L3, L4, L5)

UNIT-1: Linear Algebra Foundations
Vector spaces, norms, inner products, orthogonality, and projections o Eigenvalues/eigenvectors, spectral theorem, and invariant subspaces, Vector spaces, norms, inner products, orthogonality, and projections o Eigenvalues/eigenvectors, spectral theorem, and invariant subspaces
UNIT-II: Optimization for Learning
Unconstrained methods: gradient descent, line search, Newton/quasi- Newton o Constrained optimization: Lagrange multipliers, projected and barrier methods o Convex sets/functions, KKT conditions for nonlinear programing problems. o Regularization: l1/l2penalties, bias–variance trade- offs in risk minimization o Convergence rates and step- size strategies in deterministic models.
UNIT-III: Probability and Statistical Learning
Random variables, expectations, covariance; exponential family basics o Bayesian inference: conjugacy, MAP vs. MLE, posterior predictive analysis o Hypothesis testing and confidence intervals for model comparison o Information measures: entropy, KL divergence, mutual information in learning o Generalization, overfitting, and model selection criteria(e.g., AIC/BIC/VC-style capacity). o Concentration and uncertainty quantification for predictions
UNIT-IV: Models and Algorithms
Linear and kernel methods: least squares, logistic regression, and SVMs o Dimensionality reduction: PCA, kernel PCA, and manifold intuition o Probabilistic models: Naive Bayes, Gaussian mixtures, EM overview o Neural network mathematics: backpropagation, initialization, normalization o Regularization schemes: weight decay, early stopping, dropout perspectives
UNIT-V: Integrative Applications and Worked Examples
End- to- end ML pipelines: preprocessing, scaling/whitening, and feature engineering o PCA- based exploration and anomaly detection in high- dimensional data o SVMs and convex models for fault diagnosis and predictive maintenance o Bayesian A/B testing and decision- making under uncertainty o Neural network design trade- offs: capacity, optimization, and generalization o Model evaluation: calibration, ROC/PR analysis, and uncertainty reporting

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Textbooks

1. Deisenroth, M. P., Faisal, A. A., & Ong, C. S. Mathematics for Machine Learning. Cambridge University Press, 2020
2. Murphy, K. P. Probabilistic Machine Learning: An Introduction. MIT Press, 2022
3. Boyd, S., & Vandenberghe, L. Convex Optimization. Cambridge University Press, 2004

References

1. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016
2. Shalev- Shwartz, S., & Ben- David, S. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014
3. Jolliffe, I. T., & Cadima, J. Principal component analysis: a review and recent developments. Phil. Trans. R. Soc. A, 2016
4. Cover, T. M., & Thomas, J. A. Elements of Information Theory, 2nd ed. Wiley, 2006
5. Murphy, K. P. Probabilistic Machine Learning: Advanced Topics. MIT Press, 2023.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

STATISTICAL LEARNING THEORY AND MATHEMATICAL FOUNDATIONS OF AI

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3DOE301c	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

1. Build rigorous probabilistic, information-theoretic, and optimization foundations that support modern AI across engineering domains.
2. Unify theory and algorithms to enable principled design and analysis of learning systems for cross- disciplinary problems.
3. Connect generalization theory with practice through concise formulations and proofs suited to PG research contexts.

Course Outcomes (COs)

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

CO1: Analyze probabilistic models, convergence theorems, and stochastic processes underlying learning Algorithms (L3, L4).

CO2: Formulate and prove generalization bounds using VC dimension, Rademacher complexity, and stability (L4, L5).

CO3: Derive and optimize loss/regularization for linear, kernel, and deep models; justify selections by convexity and smoothness properties (L4, L5).

CO4: Evaluate algorithms via bias–variance, PAC guarantees, and information- theoretic criteria such as KL and mutual information (L4, L5).

CO5: Integrate theory to design robust AI solutions for multi- domain engineering applications and communicate findings effectively (L5).

UNIT-1: Probability and Measure Foundations
Probability spaces, σ- algebras, random variables; expectation and conditional expectation. • Inequalities and concentration: Markov, Chebyshev, Hoeffding; LLN and CLT. • Modes of convergence and Borel–Cantelli; almost sure vs in- probability convergence. • Random processes and martingale basics; optional stopping; Doob's inequality (overview). • Markov chains: ergodicity, mixing, stationary distributions for modeling sequences.
UNIT-II: Statistical Learning Theory
Probability spaces, σ- algebras, random variables; expectation and conditional expectation. • Inequalities and concentration: Markov, Chebyshev, Hoeffding; LLN and CLT. • Modes of convergence and Borel–Cantelli; almost sure vs in- probability convergence. • Random processes and martingale basics; optional stopping; Doob's inequality (overview). • Markov chains: ergodicity, mixing, stationary distributions for modeling sequences.
UNIT-III: Optimization for Learning
Convex analysis: Lipschitzness, smoothness, strong convexity; implications for rates. • Gradient, stochastic, and variance- reduced methods; step- size and convergence trade- offs. • Proximal methods and projected gradients; sparsity via ℓ1 (Lasso) and shrinkage via ℓ2 (ridge). • Duality and KKT conditions; constrained learning formulations. • Nonconvex landscapes in deep networks—saddle points, over-parameterization, and implicit regularization. • Generalization–optimization interplay: implicit bias of optimizers.
UNIT-IV: Kernel and Probabilistic Models
Reproducing kernel Hilbert spaces, kernel trick, representer theorem. • Large- margin methods: SVMs, soft margins, hinge loss; primal–dual views. • Gaussian processes: kernels as priors; posterior prediction and uncertainty quantification. • Graphical models—Bayesian networks and Markov/conditional random fields for structured prediction; exact inference (variable elimination, junction tree) and approximate methods (loopy belief propagation, variational). • Hidden Markov models; EM for latent- variable learning. • Variational inference and message passing for scalable probabilistic AI.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UNIT-V: Deep Learning and Information Theory
Backpropagation (chain rule), initialization, normalization, and activation design. • Loss functions and calibration; cross- entropy, margin losses, and robust objectives. • Generalization in deep nets: margins, flat minima, compression, and stability views. • Information- theoretic tools: entropy, mutual information, KL divergence. • Information bottleneck and representation learning, with links to PAC- Bayes. • Worked examples: applying theory to vision, language, and control tasks in engineering.

Textbooks:

1. Shalev- Shwartz, S., Ben- David, S., Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, first edition, 2014.
2. Goodfellow, I., Bengio, Y., Courville, A., Deep Learning, MIT Press, 2016.
3. Bishop, C. M., Pattern Recognition and Machine Learning, Springer, 2006.

References:

1. Vapnik, V. N., The Nature of Statistical Learning Theory, Springer, 1995/1998.
2. Cover, T. M., Thomas, J. A., Elements of Information Theory, 2nd ed., Wiley, 2006.
3. Mohri, M., Rostamizadeh, A., Talwalkar, A., Foundations of Machine Learning, 2nd ed., MIT Press, 2018.
4. Schölkopf, B., Smola, A. J., Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.
5. Rasmussen, C. E., Williams, C. K. I., Gaussian Processes for Machine Learning, MIT Press, 2006.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHEMISTRY OF NANOMATERIALS AND APPLICATIONS IN ENGINEERING

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3D93203b	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

1. Understand the fundamentals of nanoscience and nanotechnology.
2. Learn different methods for synthesis and characterization of nanomaterials.
3. Study the chemical properties and surface chemistry of nanomaterials.
4. Analyze the physical and mechanical properties of nanostructured materials.
5. Explore engineering and industrial applications of nanomaterials.

Course Outcomes (COs)

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

CO1: Explain the basics, classification, importance, and natural occurrence of nanomaterials.
CO2: Apply suitable top-down and bottom-up synthesis methods to prepare nanomaterials.
CO3: Analyze nanomaterials using diffraction, spectroscopy, microscopy, and particle size techniques.
CO4: Evaluate the properties and applications of carbon-based, magnetic, and functional nanomaterials.
CO5: Design and implement nanomaterial-based solutions for engineering, medical, and environmental applications

UNIT-1
Basics of Nanomaterials:Introduction, Scope of nanoscience and nanotechnology, nanoscience in nature, classification of nanostructured materials, importance of nanomaterials.
UNIT-II
Synthesis of nanomaterials: Top-Down approach,Inert gas condensation, arc discharge method, aerosol synthesis, plasma arc technique, ion sputtering, laser ablation, laser pyrolysis, and chemical vapor deposition method, electrode position method, high-energy ball milling method. Synthetic Methods: Bottom-Up approach, Sol-gel synthesis, microemulsions or reverse micelles, co precipitation method, solvothermal synthesis, hydrothermal synthesis, microwave heating synthesis and sonochemical synthesis.
UNIT-III
Techniques for characterization: Diffraction techniques, spectroscopy techniques, electron microscopy techniques for the characterization of nanomaterials, BET method for surface area analysis, dynamic light scattering for particle size determination.
UNIT-IV
Studies of Nano-structured Materials: Synthesis, properties and applications of the following nanomaterials: fullerenes, carbon nanotubes, 2D-nanomaterial (Graphene), core-shell, magnetic nanoparticles, thermoelectric materials, non-linear optical materials.
UNIT-V
Advanced Engineering Applications of Nanomaterials: Applications of nanoparticles, nanorods, nano wires inWater treatment, sensors, electronic devices, medical domain, civil engineering, chemical engineering, metallurgy and mechanical engineering, food science, agriculture, pollutants degradation.

TEXT BOOKS:

1. NANO: The Essentials:T Pradeep, MaGraw-Hill, 2007.
2. Textbook of Nanoscience and nanotechnology: B S Murty, P Shankar, Baldev Rai, BB Rath
and James Murday, Univ. Press, 2012.

REFERENCE BOOKS:

1. Concepts of Nano chemistry; LudovicoCademrtiri and Geoffrey A. Ozin& Geoffrey A. Ozin, Wiley-VCH, 2011.
2. Nanostructures &Nanomaterials; Synthesis, Properties & Applications: Guozhong Cao, Imperial College Press, 2007.

St. Johns College of Engineering & Technology

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

3. Nanomaterials Chemistry, C. N. R. Rao, Achim Muller, K.Cheetham, Wiley-VCH, 2007.

PHOTONICS FOR ENGINEERS

I M. Tech – III Semester								SJCET-R25
Course Code	Category	Hours/Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
25G3DOE301e	OE-1	3	0	0	3	40	60	100

Course Objectives (COs)

- Understand the fundamental nature of light, wave propagation, and basic optical phenomena such as interference, diffraction, and polarization.
- Learn the principles and working of photonic devices such as LEDs, laser diodes, and photodetectors.
- Study nonlinear optical effects and modulation techniques used in modern photonic systems.
- Analyze optical waveguides and fiber-optic communication systems for data transmission applications.
- Explore advanced photonic systems, sensors, quantum photonics, and emerging technologies such as plasmonics and terahertz photonics.

Course Outcomes (COs)

The objectives of this course are to:

CO 1: gain the ability to describe how light behaves and propagates in optical media.

CO 2: explain light–matter interaction mechanisms and analyze the working, characteristics, and applications of LEDs, laser diodes, quantum well lasers and various photodetectors.

CO 3: explain key nonlinear effects and analyze the operation of optical modulators and switches such as the Mach–Zehnder Interferometer.

CO 4: analyze fiber parameters and explain the functioning of transmitters, receivers, WDM systems, couplers and resonator-based communication devices.

CO 5: describe the working of photonic sensors and emerging quantum and ultrafast photonic technologies used in sensing, computation and communication.

UNIT-1
Unit I – Fundamentals of Photonics: Nature of light: wave-particle duality, polarization, coherence. Maxwell's equations and wave propagation in dielectric media. Reflection, refraction, (vector notation) interference, diffraction, dispersion and birefringence.
UNIT-II
Unit II – Photonic Devices and Components – I :Absorption, spontaneous and stimulated emission. Einstein coefficients, population inversion, optical gain. Semiconductor light sources: LEDs, laser diodes, quantum well lasers. Photodetectors: PIN, avalanche photodiodes, photomultiplier tubes.
UNIT-III
Unit III – Photonic Devices and Components – II :Nonlinear optical effects – second-harmonic generation, Kerr effect, four-wave mixing. Electro-optic Magneto optic and acousto-optic modulation principles. Optical modulators and switches (Mach ZehnderInterferometer).
UNIT-IV
Unit IV – Optical Waveguides and Applications: Optical fibers – numerical aperture, V-number, modes, attenuation, dispersion.Fiber-optic communication systems: transmitters, receivers, multiplexing (WDM), optical couplers, ring resonators.
UNIT-V
Unit V – Photonic Systems and Applications: Photonic sensors – interferometric, fiber Bragg gratings, biosensing. Optical signal processing and computing. Introduction to quantum photonics: single-photon sources, entanglement, and photonic qubits.Plasmonics and metamaterials, Ultrafast and terahertz photonics.

Textbooks:

1. B.E.A. Saleh& M.C. Teich — Fundamentals of Photonics (Wiley)
2. J. Singh — Semiconductor Optoelectronics: Physics and Technology
3. A. Yariv& P. Yeh — Photonics: Optical Electronics in Modern Communications
4. G. Keiser — Optical Fiber Communications

References:

1. J. Wilson & J.F.B. Hawkes — Optoelectronics: An Introduction
2. R.W. Boyd — Nonlinear Optics

3. Electromagnetic waves and radiating systems ------ E. Jorda