THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015

M E. DEGREE PROGRAMMES

CREDIT DISTRIBUTION

(For the students admitted from the Academic Year 2026-27 onwards)

Sl. No.	Category	Credits
A.	Foundation Course (FC)	03
B.	Program Core Courses (PCC)	22
C.	Program Elective Courses (PEC)	18
D.	Research Methodology and IPR course (RMC)	03
E.	Employability Enhancement Course (EEC)	27
F.	Value Added Courses (VAC) (Non-Credit and not included in CGPA) (04 credits)	-
	Minimum Credits to be earned for the award of the Degree	73
		From A to E and the successful completion of F

Draft - SCHEDULING OF COURSES FOR STUDENTS JOINED FROM ACADEMIC YEAR 2026-27 ONWARDS

Sem	Theory						Theory cum Practical	Practical	Value Added Courses	Employability Enhancement Courses	Credit
	1	2	3	4	5	6	7	8			
I	**26PDFA0 Performance Modeling** (Foundation Course - 3 credits)	**26PDCA0 Algorithm Design and Analysis** (PCC - 3 credits)	**26PDCB0 Parallel and Distributed Computing** (PCC - 3 credits)	**26PDCC0 Applied Cryptography** (PCC - 3 credits)		**26PDPA0 Data Modelling and Visualization** (PEC-1 - 3 credits)	**26PDCD0 Agile Software Development (TCP)** (PCC - 3 credits)	**26PDCE0 Algorithm Design and Analysis Lab** (PCC - 2 credits)	VAC 1		20
II	**26PDCF0 Cloud Computing Systems and Services** (PCC - 3 credits)	PEC-2 3 credits	PEC-3 3 credits			**Research Methodology and IPR** 3 Credits	**26PDCG0 Deep Learning** (PCC - 3 credits)	**26PDCH0 Systems Programming Lab** (PCC - 2 credits)	VAC -2	Mini Project 3 credits	20
III	PEC-4 3 credits	PEC-5 3 credits	PEC-6 3 credits							Project-I 9 credits	18
IV										Project-II 15 Credits	15
											73

M.E. (CSE) Categorization of Courses

A. Foundation Course: (3 Credits)

- 26PDFA0 – Performance Modeling (3 Credits)

B. Programme Core Courses: (22 Credits)

Theory:

- 26PDCA0 - Algorithm Design and Analysis (3 Credits)
- 26PDCB0 - Parallel and Distributed Computing (3 Credits)
- 26PDCC0 - Applied Cryptography (3 Credits)
- 26PDCF0 - Cloud Computing Systems and Services (3 Credits)

Theory cum Practical:

- 26PDCD0 - Agile Software Development (3 Credits)
- 26PDCG0 - Deep Learning (3 Credits)

Practical:

- 26PDCE0 - Algorithm Design and Analysis Lab (2 Credits)
- 26PDCH0 - Systems Programming Lab (2 Credits)

C. Elective Courses: (18 Credits)

- 22PDPA0 - Data Modelling and Visualization (I SEM) - 3 Credits
- Applied Machine Learning
- Randomized Algorithms
- Quantum Computing
- GPU Programming

- Deep Learning for NLP
- Deep Learning for Vision
- Probabilistic Graphical Modelling
- Computer Security and Forensics
- Formal Verification and Model Checking
- Design and Analysis of Parallel Algorithms
- Network Performance and Vulnerability Analysis
- Data Center design and Management
- Protocol Design and Verification
- Computational Geometry
- Fault Tolerant Computer Systems

D. Research Methodology and IPR course (3 Credits)

E. Employability Enhancement Course (27 Credits)

- Mini Project (3 Credits)
- Project-I (9 Credits)
- Project-II (15 Credits)

M.E. COMPUTER SCIENCE AND ENGINEERING DEGREE PROGRAMME

CURRICULUM AND DETAILED SYLLABI

FOR

FIRST SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2026-27 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
M.E. Degree Programme

COURSES OF STUDY
(For the candidates admitted from 2026-27 onwards)

FIRST SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
26PDFA0	Performance Modeling	FC	3	-	-	3
26PDCA0	Algorithm Design and Analysis	PCC	3	-	-	3
26PDCB0	Parallel and Distributed Computing	PCC	3	-	-	3
26PDCC0	Applied Cryptography	PCC	3	-	-	3
PROGRAMME ELECTIVE						
26PDPA0	Data Modelling and Visualization	PEC	3	-	-	3
THEORY CUM PRACTICAL						
26PDCD0	Agile Software Development	PCC	2	-	2	3
PRACTICAL						
26PDCE0	Algorithm Design and Analysis Lab	PCC	-	-	4	2
		Total				**20**

FC : Foundation Course
PCC : Programme Core Courses
PEC : Programme Elective Courses

L : Lecture
T : Tutorial
P : Practical

Note:
1 Lecture Period/week is equivalent to 1 credit
1 Tutorial Period/week is equivalent to 0.5 credit
1 Practical Period/week is equivalent to 0.5 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
B.E. / B.Tech. Degree Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2026-27 onwards)

FIRST SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	26PDFA0	Performance Modeling	3	40	60	100	27	50
2	26PDCA0	Algorithm Design and Analysis	3	40	60	100	27	50
3	26PDCB0	Parallel and Distributed Computing	3	40	60	100	27	50
4	26PDCC0	Applied Cryptography	3	40	60	100	27	50
PROGRAMME ELECTIVE								
5	26PDPA0	Data Modelling and Visualization	3	40	60	100	27	50
THEORY CUM PRACTICAL								
6	26PDCD0	Agile Software Development	3 (Terminal Exam: Theory)	50	50	100	22.5	50
PRACTICAL								
7	26PDCE0	Algorithm Design and Analysis Lab	3	60	40	100	18	50

		Category	L	T	P	Credit
26PDFA0	**PERFORMANCE MODELING**	FC	3	0	0	3

Preamble

This course provides a comprehensive understanding of performance modeling and evaluation techniques for modern computer and communication systems. The course focuses on workload characterization, queuing and stochastic models, experimental design, simulation techniques, benchmarking, and capacity planning approaches used in performance engineering. It enables students to analyze system behavior, evaluate performance metrics, and interpret performance data for distributed, cloud, and data-intensive computing environments.

Prerequisite

- Probability and Statistics
- Computer Networks
- Operating Systems

Course Outcomes

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

No.	Course Outcome	TPS Level	CO Weightage	SDG addressed	SDG Level
CO1	Examine performance metrics and evaluation techniques to assess the behaviour of computer and communication systems.	TPS3	14%	-	-
CO2	Differentiate workload characteristics and benchmarking methodologies for effective system performance assessment.	TPS4	20%	-	-
CO3	Utilize queuing and stochastic models to evaluate the performance of computing and distributed service systems.	TPS3	20%	-	-
CO4	Interpret experimental results using statistical analysis and performance visualization techniques.	TPS4	20%	-	-
CO5	Develop simulation and capacity planning models for cloud-based computing environments.	TPS3	14%	-	-
CO6	Analyze performance characteristics of computing systems through case studies involving scalability, bottlenecks, and resource utilization.	TPS4	12%	-	-

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1.	S	M	L								L

CO2.	S	S	M	M					L		L
CO3.	S	M	M	L	L						
CO4.	S	S	M	M							L
CO5.	S	M	M	L	L				L		L
CO6.	S	S	M	M	L			L			L

Assessment Pattern

CO	CAT1	Assignment 1	CAT2	Assignment 2	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	30	40	-	-	14
CO2	35	35	-	-	20
CO3	35	25	-	-	20
CO4	-	-	50	-	20
CO5	-	-	50	-	14
CO6	-	-	-	100	12
Total	**100**	**100**	**100**	**100**	**100**

Syllabus

Module 1:

Fundamentals of Performance Modeling – Introduction to Performance Engineering – Performance Metrics – Selection of Performance Metrics – System Bottlenecks – Scalability Concepts – Throughput, Response Time and Utilization – Performance Evaluation Techniques – Analytical, Simulation and Measurement-Based Approaches – Monitoring Techniques and Observability – Common Errors in Performance Studies.

Module 2:

Workload Characterization and Benchmarking – Workload Types and Characteristics – Statistical Methods for Workload Analysis – Discrete and Continuous Probability Distributions – Memoryless Property – Averaging and Histograms – Clustering Techniques – Principal Component Analysis – Profiling Techniques – Benchmarking Methodologies – SPEC, TPC and MLPerf Benchmarks.

Module 3:

Queuing and Stochastic Models – Introduction to Stochastic Processes – Birth-Death Processes – Introduction to Queuing Theory – M/M/1 and M/G/1 Queues – Multi-Server Queue Models – Markov Chains – Queue Networks – Operational Laws and Applications – Performance Analysis of Distributed Service Systems.

Module 4:

Experimental Design and Performance Analysis – Experimental Design Techniques – Full Factorial Design – Fractional Factorial Design – Analysis of Variance (ANOVA) – Test of Significance – Performance Data Collection – Statistical Interpretation of Results – Gantt Charts – Kiviat Graphs – Performance Visualization Techniques.

Module 5:

Simulation and Capacity Planning – Monte Carlo Simulation – Discrete Event Simulation – Simulation Verification and Validation – Random Number Generation – Capacity Planning Techniques – Resource Provisioning – Cloud SLA Modeling – Bottleneck Analysis and System Scalability.

Module 6:

Performance Evaluation of Computing Systems – Performance Analysis of Communication Networks – Cloud Computing Systems –Database Systems –AI and Data-Intensive Applications – Case Studies on Scalability and Bottleneck Analysis.

6. Learning Resources

6.1 Text Book

1. Raj Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling", Wiley-India, Reprint Edition, 2015.
2. Kishor S. Trivedi, "Probability and Statistics with Reliability, Queueing, and Computer Science Applications", 2nd Edition, Wiley India, 2016.
3. Brendan Gregg, "Systems Performance: Enterprise and the Cloud", 2nd Edition, Pearson Education, 2021.

6.2 Reference Books & Web resources

1. André B. Bondi, "Foundations of Software and System Performance Engineering Process, Performance Modeling, Requirements, Testing, Scalability, and Practice", Addison-Wesley Professional, 2015.
2. Mor Harchol-Balter, Performance Modeling and Design of Computer Systems: Queueing Theory in Action, Cambridge University Press, 2013
3. Gunter Bolch, Stefan Greiner, Hermann de Meer, Kishor S. Trivedi," Queueing Networks and Markov Chains", 2nd Edition, Wiley-Interscience, 2006.
4. Maarten van Steen, Andrew S Tanenbaum, "Distributed Systems", 3rd Edition, Pearson, 2023.
5. Nptel Course: "Performance Evaluation of Computer Systems, Prof. Krishna Moorthy Sivalingam" IIT Madras - https://nptel.ac.in/courses/106106048

7. Course Contents and Lecture Schedule

Module No	Topics	No. of Periods
1	**Fundamentals of Performance Modeling**	**6**
	Introduction to Performance Engineering and Performance Metrics	1
	System Bottlenecks	1
	Throughput, Response Time and Utilization	1
	Performance Evaluation Techniques	1
	Analytical, Simulation and Measurement-Based Approaches –	1

	Monitoring Techniques and Observability and Common Errors in Performance Studies	1
2	**Workload Characterization and Benchmarking**	**6**
	Workload Types and Characteristics	1
	Statistical Methods for Workload Analysis	1
	Averaging and Histograms	1
	Clustering Techniques and Principal Component Analysis	1
	Profiling Techniques and Benchmarking Methodologies	2
3	**Queuing and Stochastic Models**	**6**
	Introduction to Queuing Theory and Birth-Death Processes	1
	M/M/1 and M/G/1 Queue Models	1
	Multi-Server Queue Models	1
	Markov Chains and Queue Networks	1
	Operational Laws and Applications	1
	Performance Analysis of Distributed Service Systems	1
4	**Experimental Design and Performance Analysis**	**6**
	Experimental Design Techniques	1
	Factorial Design and Analysis of Variance (ANOVA)	2
	Test of Significance and Performance Data Collection	1
	Statistical Interpretation of Results	1
	Gantt Charts – Kiviat Graphs – Performance Visualization Techniques	1
5	**Simulation and Capacity Planning**	**6**
	Monte Carlo and Discrete Event Simulation	1
	Simulation Verification and Validation	1
	Random Number Generation and Capacity Planning Techniques	1
	Resource Provisioning	1
	Cloud SLA Modeling	1
	Bottleneck Analysis and System Scalability.	1

6	**Performance Evaluation**	**6**
	Performance Analysis of Communication Networks	1
	Cloud Computing Systems	1
	Database Systems	1
	AI and Data-Intensive Applications	1
	Predictive Performance Analytics	1
	Case Study on Scalability and Bottleneck Analysis	1
	Total Periods	**36**

8.Course Designer(s):

1. Mr.D.Nagendra Kumar, Asst. Prof., CSE, dnkcse@tce.edu
2. Ms.C.Karthika, Asst. Prof., CSE, ckrcse@tce.edu

26PDCA0	ALGORITHM DESIGN AND ANALYSIS	Category	L	T	P	Credit
		PCC	3	0	0	3

1. Preamble

This course equips students with a rigorous, graduate-level foundation in algorithm design and analysis. Building on UG-level knowledge of sorting, searching, and basic graph algorithms, the course advances into asymptotic theory, advanced dynamic programming, network flow, computational complexity, approximation algorithms, fixed-parameter tractability, and randomized methods.

2. Prerequisite

- Data Structures and Algorithms

3. Course Outcomes

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

No.	Course Outcome	TPS Level	CO Weightage	SDG addressed	SDG Level
CO1	Apply asymptotic notations (O, Ω, Θ, o, ω) and complexity analysis techniques, solve recurrences via the Akra-Bazzi method, and apply amortized analysis techniques.	TPS3	15%	SDG 4	1
CO2	Analyze the different algorithmic paradigms such as dynamic programming and advanced greedy strategies to solve optimization problems.	TPS4	20%	SDG 9	2
CO3	Analyze the various graph algorithms and linear programming to solve combinatorial problems.	TPS4	20%	SDG 9	2
CO4	Demonstrate complexity classes such as P, NP, and NP-Complete along with heuristic approaches for solving difficult computational problems.	TPS3	15%	SDG 9	1
CO5	Apply approximation algorithms with proven ratios and apply fixed-parameter tractable algorithms and kernelization for parameterized problems.	TPS3	12%	SDG 4	1
CO6	Apply randomized algorithm techniques — Las Vegas, Monte	TPS3	18%	SDG 9	2

	Carlo, randomized selection, Skyline Matrix methods, and primality testing — with rigorous probabilistic analysis.				

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1.	M	M	S	L				M	M	L	L
CO2.	S	S	M	S				M	M	L	L
CO3.	S	S	M	S	M	M		M	M	L	L
CO4.	M	M	S	L	M			M	M	L	L
CO5.	M	M	S	L				M	M	L	L
CO6.	M	M	S	L	M			M	M	L	L

4. Assessment Pattern

CO	CAT1	Assignment 1	CAT2	Assignment 2	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	20	30			15
CO2	40	30			20
CO3	40	40			20
CO4			30	30	15
CO5			30	30	15
CO6			40	40	15
Total	100	100	100	100	100

5. Syllabus

Advanced Analysis of Algorithms

Introduction – Asymptotic Notations - Formal definitions and properties of O, Ω, Θ, o (small-o), and ω (small-omega). Limit-based comparisons between functions; transitivity, symmetry, reflexivity of asymptotic relations. Distinguishing o from O; proof techniques and common misconceptions.

Recurrence Relations — Advanced Techniques: Substitution method with tight bounds; recursion tree method; Master theorem (all three cases with proof sketch).

Amortized Analysis: Aggregate method, accounting method, potential method.

Advanced Dynamic Programming and Greedy Algorithms

String and Sequence DP: Longest Common Subsequence, Edit Distance (Levenshtein), Sequence Alignment (Needleman-Wunsch); suffix arrays (construction and use); applications in computational biology.

Advanced Greedy Algorithms: Weighted job scheduling; Huffman codes optimality proof.

Advanced Graph Algorithms and Linear Programming

Advanced Graph Algorithms: Introduction - Strongly connected components — Kosaraju's algorithms; All-pairs shortest paths — Floyd-Warshall algorithm. Maximum Flow Algorithms - Ford-Fulkerson method and correctness and Edmonds-Karp.

LP in Combinatorial Optimization (Overview): LP formulation of combinatorial problems; duality and complementary slackness; LP relaxations and integrality gaps — applications in vertex cover and matching.

Complexity Theory and Intractability

Complexity Classes: Class P; Class NP — certificate and NTM definitions, equivalence; Class co-NP; PSPACE and Savitch's theorem.

NP-Completeness and Reductions: Polynomial-time Karp reductions; Cook-Levin theorem (SAT is NP-complete — proof sketch); key NP-complete reductions: 3-SAT, Clique, Hamiltonian Cycle, TSP.

Coping with Intractability: Exact exponential algorithms — branch and bound with pruning.

Approximation Algorithms and Fixed-Parameter Tractability

Approximation Algorithm Design: Approximation ratio — definition and examples; PTAS and FPTAS definitions; Greedy approximation: Vertex Cover 2-approximation; Set Cover O(log n)-approximation; scheduling on parallel machines; LP-rounding for weighted vertex cover.

Parameterized problems; FPT and XP classes; kernelization and bounded search tree algorithm for Vertex Cover.

Randomized Algorithm Techniques

Probability Foundations for Algorithm Analysis: Expectation, linearity of expectation;

Las Vegas and Monte Carlo Algorithms: Las Vegas algorithms — always correct, random runtime: Randomized QuickSort (expected O(n log n) analysis), Monte Carlo algorithms — always fast, bounded error; one-sided vs two-sided error; Karger's min-cut; Skyline Matrix method — definition, properties, and application in randomized algorithms.

Randomized Selection: Randomized selection (RandSelect) — expected O(n) for k-th order statistic; analysis via indicator random variables; comparison with deterministic Median-of-Medians O(n) selection.

6. Learning Resources

6.1 Text Book

1. Dasgupta, Papadimitrou and Vazirani, "Algorithms", McGraw-Hill Education, 2023.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", MIT Press, fourth edition, 2022.

6.2 Reference Books & Web resources

1. Jonathan Katz and Yehuda Lindell, "Introduction to Modern Cryptography", Chapman & Hall, Second Edition, 2014.
2. Joseph Jaja, "An introduction to Parallel Algorithms", Addison-Wesley, 1992.
3. Juraj Hromkovic– Design and Analysis of Randomized Algorithms, First edition, Springer, 2005

7. Course Contents and Lecture Schedule

Module No	Topics	No of Periods
1.	**Advanced Analysis of Algorithms**	
1.1	Introduction – Asymptotic Notations - Formal definitions and properties of O, Ω, Θ, o (small-o), and ω (small-omega). Limit-based comparisons between functions; transitivity, symmetry, reflexivity of asymptotic relations. Distinguishing o from O; proof techniques and common misconceptions.	2
1.2	Recurrence Relations — Advanced Techniques: Substitution method with tight bounds; recursion tree method; Master theorem (all three cases with proof sketch).	2
1.3	Amortized Analysis: Aggregate method, accounting method, potential method.	2
2.	**Advanced Dynamic Programming and Greedy Algorithms**	
2.1	String and Sequence DP: Longest Common Subsequence, Edit Distance (Levenshtein).	2
2.2	Sequence Alignment (Needleman-Wunsch); suffix arrays (construction and use); applications in computational biology.	2

2.3	Advanced Greedy Algorithms: Weighted job scheduling; Huffman codes optimality proof.	2
3.	**Advanced Graph Algorithms and Linear Programming**	
3.1	Advanced Graph Algorithms: Introduction - Strongly connected components — Kosaraju's algorithms; All-pairs shortest paths — Floyd-Warshall algorithm.	2
3.2	Maximum Flow Algorithms - Ford-Fulkerson method and correctness and Edmonds-Karp.	2
3.3	LP in Combinatorial Optimization (Overview): LP formulation of combinatorial problems; duality and complementary slackness; LP relaxations and integrality gaps — applications in vertex cover and matching.	2
4.	**Complexity Theory and Intractability**	
4.1	Complexity Classes: Class P; Class NP — certificate and NTM definitions, equivalence; Class co-NP; PSPACE and Savitch's theorem.	2
4.2	NP-Completeness and Reductions: Polynomial-time Karp reductions; Cook-Levin theorem (SAT is NP-complete — proof sketch); key NP-complete reductions: 3-SAT, Clique, Hamiltonian Cycle, TSP.	2
4.3	Coping with Intractability: Exact exponential algorithms — branch and bound with pruning.	2
5	**Approximation Algorithms and Fixed-Parameter Tractability**	
5.1	Approximation Algorithm Design: Approximation ratio — definition and examples; PTAS and FPTAS definitions; Greedy approximation: Vertex Cover 2-approximation;	2
5.2	Set Cover O(log n)-approximation; scheduling on parallel machines; LP-rounding for weighted vertex cover.	2
5.3	Parameterized problems; FPT and XP classes; kernelization and bounded search tree algorithm for Vertex Cover.	2
6	**Randomized Algorithm Techniques**	

6.1	Probability Foundations for Algorithm Analysis: Expectation, linearity of expectation	1
6.2	Las Vegas and Monte Carlo Algorithms: Las Vegas algorithms — always correct, random runtime: Randomized QuickSort (expected O(n log n) analysis),	2
6.3	Monte Carlo algorithms — always fast, bounded error; one-sided vs two-sided error; Karger's min-cut; Skyline Matrix method — definition, properties, and application in randomized algorithms.	2
6.4	Randomized Selection: Randomized selection (RandSelect) — expected O(n) for k-th order statistic; analysis via indicator random variables; comparison with deterministic Median-of-Medians O(n) selection.	1
	Total Hours	**36**

8.Course Designer(s):

1. Raja Lavanya, Assistant Professor, Department of Computer Science and Engineering — rlit@tce.edu

2. Dharani J, Assistant Professor, Department of Computer Science and Engineering — jdicse@tce.edu

26PDCB0	PARALLEL AND DISTRIBUTED COMPUTING	Category	L	T	P	Credit
		PCC	3	0	0	3

1. Preamble

This course introduces learners about the basic concepts of parallel architectures, parallel programming models, distributed system concepts and emerging paradigms. Upon completing this course, learners will possess the competencies required for high-performance computing (HPC), distributed systems research, cloud infrastructure design, and AI/ML workload parallelisation

2. Prerequisite

Computer Architecture

Operating Systems

Computer Network

3. Course Outcomes

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

No.	Course Outcome	TPS Level	CO Weightage	SDG addressed	SDG Level
CO1	Use parallel architecture classifications and performance metrics to solve problems related to the speedup, efficiency, and scalability of parallel computing systems	TPS3	15	-	-
CO2	Apply efficient parallel solutions for canonical computational problems by implementing design strategies	TPS3	15	-	-
CO3	Implement reliable and coherent distributed applications by employing core principles such as clock synchronization, mutual exclusion, election algorithms,	TPS3	20	-	-
CO4	Use distributed deadlock detection, and consistency models, File System models that reveals concurrent programming errors	TPS3	15	-	-
CO5	Implement high-performance parallel programs for distributed and heterogeneous platforms using MPI and the CUDA GPU programming model	TPS3	15	-	-
CO6	Solve resilient distributed systems that meet availability and reliability requirements in a Big Data Framework	TPS3	20	9	2

Mapping with Programme Outcomes

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO3.	S	M	L								
CO4.	S	M	L	M							
CO3.	S	M	L	M				L			
CO4.	S	M	L	M				M			L

CO5.	S	M	L	M	L						L
CO6.	S	M	L	M	L						L

S- Strong; M-Medium; L-Low

4. Assessment Pattern

CO	CAT1	Assignment 1	CAT2	Assignment 2	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	30	30			15
CO2	30	30			15
CO3	40	40			20
CO4			30	30	15
CO5			30	30	15
CO6			40	40	20
Total	**100**	**100**	**100**	**100**	**100**

5. Syllabus

Fundamentals Of Parallel and Distributed Computing- Evolution from sequential to parallel computing-Distinction between parallel computing and distributed computing -Flynn's Taxonomy and Parallel Architecture Classifications-Shared Memory Architectures, Distributed Memory Architecture,HPC cluster architectures, Multicore and Many-core Processors, Interconnection Networks-Performance Metrics and Laws-Amdahl's Law, Gustafson’s Law,Karp-Flatt Metric, Work-Span Model, Roofline Model

Parallel Programming Models and Algorithms- Parallelism Models and Programming Paradigms -Process-level vs. Thread-level parallelism, Data Parallelism, Memory Consistency and Cache Coherence -Theoretical Models of Parallel Computation-PRAM (Parallel Random Access Machine) Model- Parallel Join Formulation and Cost Optimality Analysis -Load Balancing and Scheduling-Static, Dynamic load balancing, Task Scheduling – Parallel Debugging Solution: TotalView

Distributed Computing Concepts and Algorithms - Distributed System Models and Communication - Time, Ordering, and Coordination - Clock Synchronisation-Logical, Vector, Global State Recording: Snapshot Recording algorithm, Distributed Mutual Exclusion: Token, Non Token, Election Algorithms - Distributed Deadlock and Shared Memory - Distributed File Systems and Naming-

Message Passing and GPU Computing - Message Passing Interface (MPI) - Point-to-Point Communication - Collective Communication - Derived Data Types -Communicator Management - MPI Program Development - GPU Architecture and CUDA Programming -GPU Memory Hierarchy -CUDA Synchronisation-CUDA Streams

Fault Tolerance, and Big Data Paradigms - Fault Tolerance in Distributed Systems -Failure Models-Check pointing and Rollback Recovery- Redundancy Techniques-RAID-based Data Fault Tolerance-Distributed Transactions and Consensus-Distributed Commit Protocols, Consensus Problem-Raft Consensus Algorithm-Byzantine Fault Tolerance- Zero Knowledge Proof -Map Reduce and Big Data Frameworks-CAP Theorem-No SQL

Case Study - Cluster, Grid, and Cloud Computing- Shared Memory Programming with OpenMP - Hybrid Parallel Programming -MPI + OpenMP Hybrid Model,MPI + CUDA - Parallel Computing Infrastructure in India: Supercomputing & HPC, AI & GPU Compute-– Block Chain: - Green computing concepts

6. Learning Resources

6.1 Text Book

1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, 2nd Edition, McGraw-Hill Education, 2022
2. Andrew S. Tanenbaum and Maarten Van Steen, Distributed Systems: Principles and Paradigms, 4th Edition, Pearson,2023

6.2 Reference Books & Web resources

1. Peter Pacheco and Matthew Malensek, An Introduction to Parallel Programming, 2nd Edition Morgan Kaufmann / Elsevier 2021
2. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Distributed Systems: Concepts and Design, 5th Edition, Pearson Education,2023
3. Ian Foster , Designing and Building Parallel Programs, Argonne National Lab ,2021
4. CUDA Programming (NVIDIA Deep Learning Institute): GPU programming with CUDA
5. Distributed Systems (MIT OpenCourseWare 6.5840): lecture notes, papers, and Go labs
6. Nptel: https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs21 (OpenMP & MPI)

7. Course Contents and Lecture Schedule

Module no.	Topic	No. of lectures
	1.Fundamentals of Parallel and Distributed Computing	
1.1	Evolution from Sequential to Parallel Computing and Distinction between Parallel and Distributed Computing	2
1.2	Flynn's Taxonomy and Parallel Architecture Classifications	
1.3	Shared Memory Architectures, Distributed Memory Architectures, HPC Clusters, Multicore and Many-core Processors, Interconnection Networks	2
1.4	Performance Metrics and Laws – Amdahl's Law, Gustafson's Law,	2
1.5	Karp-Flatt Metric, Work-Span Model and Roofline Model	1
	2.Parallel Programming Models and Algorithms	
2.1	Parallelism Models and Programming Paradigms	1
2.2	Process-level vs Thread-level Parallelism, Data Parallelism, Memory Consistency and Cache Coherence	1
2.3	Theoretical Models of Parallel Computation	1
2.3.1	PRAM (Parallel Random Access Machine) Model-	1

2.3.2	Parallel Join Formulation and Cost Optimality Analysis	1
2.3.3	Load Balancing and Task Scheduling – Static and Dynamic	1
2.3.4	Task Scheduling – Parallel Debugging Solution: TotalView	1
3. Distributed Computing Concepts and Algorithms		
3.1	Distributed System Models and Communication	1
3.2	Time, Ordering and Coordination	
3.3	Clock Synchronisation Logical, Vector, Global State Recording: Snapshot Recording algorithm	1
3.4	Distributed Mutual Exclusion: Token, Non Token, Election Algorithms	1
3.5	Distributed Deadlock and Shared Memory	1
3.6	Distributed File Systems and Naming	1
4.Message Passing and GPU Computing		
4.1	Message Passing Interface (MPI) and Point-to-Point Communication	1
4.2	Collective Communication, Derived Data Types and Communicator Management	1
4.3	MPI Program Development	1
4.4	GPU Architecture and CUDA Programming	1
4.5	GPU Memory Hierarchy, CUDA Synchronisation and CUDA Streams	1
5.Fault Tolerance and Big Data Paradigms		
5.1	Fault Tolerance in Distributed Systems and Failure Models	1
5.2	Checkpointing and Rollback Recovery	1
5.3	Redundancy Techniques and RAID-based Data Fault Tolerance	1
5.4	Distributed Transactions, Distributed Commit Protocols and Consensus Problem	1
5.5	Raft Consensus Algorithm and Byzantine Fault Tolerance	1
5.6	Zero Knowledge Proof	1

5.7	Map Reduce and Big Data Frameworks-CAP Theorem-No SQL	1
	6. Case Study	
6.1	Cluster, Grid, and Cloud Computing	1
6.2	Shared Memory Programming with OpenMP - Hybrid Parallel Programming	1
6.3	MPI + OpenMP Hybrid Mode	1
6.4	MPI + CUDA - Parallel Computing Infrastructure in India: Supercomputing & HPC, AI & GPU Computer	1
6.5	Block Chain: - Green computing concepts	1
	Total Periods	36

8. Course Designer(s):

1. Dr.R.Leena Sri,rlsit@tce.edu
2. Dr.K.Indira, kiit@tce.edu

		Category	L	T	P	Credit
26PDCC0	**APPLIED CRYPTOGRAPHY**	PCC	3	0	0	3

1. Preamble

Cryptography is the science of information and communication security. This course will discuss common security weaknesses, vulnerabilities, attack methods and mitigation approaches. The course deals with the construction and cryptanalysis of block ciphers, stream ciphers and hash functions.

2. Prerequisite

Basics of Number Theory and Algorithms

3. Course Outcomes

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

No.	Course Outcome	TPS Level	CO Weightage	SDG addressed	SDG Level
CO1	Demonstrate the fundamental theory and mathematical foundations of cryptography	TPS3	10%		
CO2	Apply symmetric encryption and decryption techniques to ensure data confidentiality.	TPS3	20%	SDG 9	2
CO3	Apply asymmetric cryptographic techniques to secure data communication by ensuring confidentiality, authentication, and data integrity.	TPS3	20%	SDG 9	2
CO4	Demonstrate cryptographic hash and message authentication techniques to ensure data integrity	TPS3	20%	SDG 9	2
CO5	Analyze cryptographic algorithms using appropriate cryptanalytic techniques to evaluate their security and identify vulnerabilities.	TPS4	20%	SDG9	2
CO6	Apply cryptographic principles and security engineering practices to design and secure software systems, APIs, cloud environments, and web services.	TPS3	10%	SDG 9	2

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	S	M	L					L	L	L	L
CO2	S	M	L					L	L	L	L
CO3	S	M	L					L	L	L	L

CO4	S	M	L					L	L	L	L
CO5	S	S	M	L				L	L	L	L
CO6	S	M	L					L	L	L	L

S- Strong; M-Medium; L-Low

4. Assessment Pattern

CO	CAT1	Assignment 1	CAT2	Assignment 2	Terminal Exam
	% of Marks	**% of Marks**	**% of Marks**	**% of Marks**	**% of Marks**
CO1	20	20			10
CO2	40	40			20
CO3	40	40			20
CO4	-	-	40	40	20
CO5	-	-	30	30	20
CO6	-	-	30	30	10
Total	100	100	100	100	100

5. Syllabus

Mathematical & Security Foundations - Overview of Cryptography and Cryptanalysis, OSI Security Architecture, Shannon's Theory of Secrecy Systems, Algebraic Structure: Groups, Rings, and Finite Fields ($GF(2^n)$), Number theory for cryptography:modular arithmetic, Euclidean algorithm, Euler's Theorem, Primality testing - Fermat's Theorem,Miller–Rabin. Classical cryptosystems, Block cipher design principles, Feistel networks and SPN structures. **Symmetric Key Design and Cryptography -** AES: design principles and operations, Modes of operation (ECB, CBC, CFB, OFB, CTR), Stream ciphers design principles- Pseudorandom Number Generators - Linear Feedback Shift Registers (LFSR), PRNGs, CSPRNGs, Statistical tests for randomness. **Public Key Cryptography & Key Management -** RSA cryptosystem, Attacks on RSA (small exponent, timing, padding issues),Diffie–Hellman key exchange, Discrete Logarithm Problem (DLP),Elliptic Curve Cryptography (ECC),Key management: generation, distribution, Needham–Schroeder protocol. **Hashing and Digital Signature:**Cryptographic hash functions and properties, Merkle–Damgård construction, Message Authentication Codes (HMAC, CMAC), Digital Signature. **Advanced Cryptanalysis:** Linear and Differential cryptanalysis foundations. Meet-in-the-middle, Time-Memory Tradeoff (Rainbow tables). Birthday attacks on collision resistance.Physical & Architectural Attacks: Side-channel attacks (Differential Power Analysis Cache-timing attacks). Adversarial Security Models: Indistinguishability under Chosen Plaintext Attack (IND-CPA) and Chosen Ciphertext Attack (IND-CCA2). **Cryptographic Applications and Security Engineering** - Secure Software Engineering (SSDLC, Threat Modeling, Secure Coding Practices), API Security (OAuth 2.0, OpenID Connect, JWT Security), Cloud Security and Cryptographic Key Management, Web Services Security, TLS 1.3. Identity-Based Encryption, Homomorphic encryption, Zero-Knowledge Proofs (zk-SNARKs, zk-STARKs).Case Study: TLS/SSL, blockchain: Hash chains, Merkle trees, End-to-End Encrypted Messaging:Forward secrecy, double ratchet algorithm.

6. Learning Resources
6.1 Text Book

1. Jonathan Katz, Yehuda Lindell, "Introduction to Modern Cryptography", 3rd Edition, CRC Press and Taylor & Francis,2025

6.2 Reference Books&Web resources

1. W. Stallings, *“Cryptography and Network Security: Principles and Practice”*, 8th Edition, Pearson Education, 2022.
2. William Easttom, *“Modern Cryptography: Applied Mathematics for Encryption and Information Security*”, 2nd Edition, Springer, 2022.
3. ChristofPaar, Jan Pelzl, and Tim Güneysu, “*Understanding Cryptography”*, 2nd Edition, Springer, 2024.

7. Course Contents and Lecture Schedule

Module No	Topics	No of Periods
1.	**Mathematical & Security Foundations**	
1.1	Overview of Cryptography and Cryptanalysis, OSI Security Architecture, Shannon’s Theory of Secrecy Systems	1
1.2	Algebraic Structure: Groups, Rings, and Finite Fields ($GF(2^n)$)	1
1.3	Number theory for cryptography: modular arithmetic, Euclidean algorithm, Euler’s Theorem	1
1.4	Primality testing - Fermat’s Theorem, Miller–Rabin	1
1.5	Classical cryptosystems, Block cipher design principles, Feistel networks and SPN structures	1
2	**Symmetric Key Design and Cryptography**	
2.1	AES: design principles and operations	1
2.2	Modes of operation (ECB, CBC, CFB, OFB, CTR)	1
2.3	Stream ciphers design principles - Pseudorandom Number Generators	1
2.4	Linear Feedback Shift Registers (LFSR)	1
2.5	PRNGs, CSPRNGs, Statistical tests for randomness	1
3	**Public Key Cryptography & Key Management**	
3.1	RSA cryptosystem	1
3.2	Attacks on RSA (small exponent, timing, padding issues)	1
3.3	Diffie–Hellman key exchange, Discrete Logarithm Problem (DLP)	2
3.4	Elliptic Curve Cryptography (ECC),Key management: generation, distribution	2
3.5	Needham–Schroeder protocol	1
4	**Hashing and Digital Signature**	
4.1	Cryptographic hash functions and properties	1

4.2	Merkle–Damgård construction	1
4.3	Message Authentication Codes (HMAC, CMAC)	2
4.4	Digital Signature	2
5	**Advanced Cryptanalysis**	
5.1	Linear and differential cryptanalysis foundations	1
5.2	Meet-in-the-middle attacks, Time–memory tradeoff attacks(Rainbow tables)	1
5.3	Birthday attack and collision resistance	1
5.4	Physical & Architectural Attacks:Side-channel attacks(Differential Power Analysis Cache-timing attacks)	1
5.5	Adversarial Security Models: Indistinguishability under Chosen Plaintext Attack (IND-CPA) and Chosen Ciphertext Attack (IND-CCA2)	2
6	**Cryptographic Applications and Security Engineering**	
6.1	Secure Software Engineering (SSDLC, Threat Modeling, Secure Coding Practices),	1
6.2	API Security (OAuth 2.0, OpenID Connect, JWT Security)	1
6.3	Cloud Security and Cryptographic Key Management	1
6.4	Web Services Security, TLS 1.3. Identity-Based Encryption	1
6.5	Homomorphic encryption, Zero-Knowledge Proofs (zk-SNARKs, zk-STARKs)	1
6.6	Case Study: TLS/SSL, blockchain: Hash chains, Merkle trees, End-to-End Encrypted Messaging:Forward secrecy, double ratchet algorithm	2
	Total	36

8.Course Designer(s):

1. Dr.M .Vijayalakshmi, Professor, CSE, mviji@tce.edu
2. Dr. N. Anita, Assistant Professor, CSE, naacse@tce.edu

		Category	L	T	P	Credit	TE-Type
26PDCD0	**AGILE SOFTWARE DEVELOPMENT**	PCC	2	0	2	3	TCP-T

Preamble

This course introduces Agile Software Development principles and practices, including Scrum, Kanban, Lean, and Extreme Programming (XP). It makes the students understand Agile planning, estimation, sprint management, team collaboration, and the use of Agile project management tools for effective software project development and management.

Prerequisite

- Software Engineering

Course Outcomes

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

No.	Course Outcome	TCE Proficiency Scale	CO Weightage	SDG Addressed	SDG Level
CO1	Explain Agile principles, Scrum, Kanban, Lean, and Extreme Programming (XP) methodologies used in software development.	TPS 2	10%		
CO2	Apply Agile project management tools and Scrum practices for software project planning and tracking.	TPS3	10%	SDG 9	2
CO3	Apply user stories, sprint planning, product backlog management, and effort estimation techniques in Agile projects.	TPS3	20%		
CO4	Analyse sprint progress using Agile metrics such as burndown charts, velocity, and cycle time.	TPS4	20%	SDG 9	2
CO5	Analyse Agile team practices and scaling Agile frameworks	TPS4	20%		
CO6	Analyse,Plan and implement a team-project for developing a complete project schedule using Project management tools like ZOHO Sprints Tool, Github, Jira and Trello	TPS 4	20%		

Assessment Pattern

CO	CAT1	CAT2	Model Test CO-wise (Cognitive, Psychomotor, Affective)	OCR/Continuous Assessment (Pre-lab, In-lab, Post-lab)	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	30		11		11
CO2	35		11	11	11
CO3	35		26	11	26
CO4	-	50	26	26	26
CO5	-	50	26	26	26
CO6	-	-	-	26	-
Total	100	100	100	100	100

Assessment Pattern: Practical Component

Type of Assessment	Component	Marks	Remarks
Continuous Assessment	Pre-lab, In-lab, Post-lab	75	Conducted throughout the semester
Model Test	CO-wise (Cognitive, Psychomotor, Affective)	25	Mid/End semester internal test
Total Internal Marks		**100**	
End Semester Examination	CO-wise (Cognitive, Psychomotor, Affective)	100	As directed by COE or as per curriculum

Syllabus

Introduction to Agile and Lean Practices:

Introduction to Agile- Why Agile? Understanding the Need for Agile Methodologies.Applying agile manifesto and principles – Agile principles for customer satisfaction, quality, teamwork – product development - Why being agile works better. Agile Approaches- Lean Software Development - Lean Principles Applied to Software Development - Value Stream Mapping: Identifying and Eliminating Waste Just-In-Time (JIT) and Pull Systems Lean Thinking in Agile Projects Extreme Programming (XP) - Overview of Extreme Programming (XP) Principles - Pair Programming - Test-Driven Development (TDD, Continuous Integration. Kanban Methodology - Introduction to Kanban: Visualizing Workflows - Kanban Boards: Creating Transparency in Your Projects - Work in Progress (WIP) Limits: Managing Flow Effectively - Continuous Delivery with Kanban.

Agile Requirements and Scrum Framework:

Functional and Non-Functional Requirements in Agile Software Development- User stories- Breaking down requirements- Estimation - Scrum Framework - Understanding Scrum: Roles, Artifacts, and Ceremonies - defining Product vision and roadmap – planning releases and sprint.

Sprint Execution and Agile Metrics:

Working through the daily scrum – Measuring Progress in Agile Projects Burndown Charts- Using Metrics for Continuous Improvement - Velocity, and Cycle Time

Agile Project Management Tools:

Introduction to Agile Tools: ZOHO Sprints, Jira, Trello, Asana - Choosing the Right Tool for

Your Team- Best Practices for Agile Tool Implementation - Integrating Agile Tools with Your Development Process- Creating a project in ZOHO Sprints.

Agile Team Dynamics:

Agile Team Dynamics - Building and Sustaining High-Performing Agile Teams - Effective Communication in Agile Environments - Dealing with Common Challenges in Agile Teams - Distributed Agile Teams: Strategies for Success.

Scaling Agile :

Challenges in Scaling Agile - Introduction to Scaled Agile Frameworks (SAFe, LeSS, Nexus, etc.). Role of AI in Agile Agile at Enterprise Level: Case Studies and Best Practices - Customizing Agile Practices for Large Organizations - Agile in Practice: Real-World Case Studies

Case Study 1: Agile Transformation in a Startup

Case Study 2: Agile Adoption in a Corporate Environment

Text Book

1. Mark C. Layton, Steven J Ostermiller, and Dean J. Kynaston, "Agile Project Management for Dummies", John Wiley & Sons, Inc., 3rd Edition, 2020

2. Mike Cohn, "Agile Estimating and Planning", Pearson, 1st Edition, 2006. (Foundational; pair with recent reference below)

3. Klaus Helbig, "Mastering Professional Scrum", Addison-Wesley, 1st Edition, 2021.

4. Steve Abrams "Agile Software Development for Beginners Mastering Flexibility and Efficiency in Modern Software Projects" 2021

Reference Books & web resources

1. Jeff Sutherland and J.J. Sutherland, "Scrum: The Art of Doing Twice the Work in Half theTime", Currency, 2014.

2. Joakim Sunden and Marcus Hammarberg, "Kanban in Action", Manning, 2014.

3. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "The DevOps Handbook", IT
4. Revolution, 2nd Edition, 2021.

5. PMI, "Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise", 2nd Edition, 2023.

6. Scrum.org resources (Scrum Guide 2020), and Atlassian Agile Guides (current online).

7. 6.Marco Tulio Valente, Software Engineering: A Modern Approach, Self- published, 2024, ISBN: 978-65-01-26467-7. Available at:https://softengbook.org

SDG Alignment

SDG 9 (Industry, Innovation and Infrastructure):

- Activity 1 (CO2, TPS 3): Compare team velocity across sprints and recommend process improvements backed by metric trends; deliverable: analysis report with charts.

- Activity 2 (CO4, TPS 4): Evaluate two Agile tools (e.g., Zoho Sprints vs Jira) against criteria for reliability, integration, and governance; deliverable: decision matrix and justification.

Course Contents and Lecture Schedule

Module No	Topics	No of Periods	**Course Outcomes**
Module 1	Introduction to Agile, Need for Agile, Agile Manifesto and Principles	2	CO1
	Agile Principles for Customer Satisfaction and Teamwork	1	CO1
	Lean Software Development, Lean Principles, Value Stream Mapping and Eliminating Waste	1	CO1
	XP Principles, Pair Programming, TDD and Continuous Integration	1	CO1
	Kanban Methodology, Kanban Boards and WIP Limits	1	CO1, CO6
Module 2	Functional & Non-Functional Requirements, User Stories and Acceptance Criteria	2	CO2, CO6
	Breaking Down Requirements, Estimation Techniques and Story Points	2	CO2, CO6
	Scrum Framework: Roles, Artifacts and Ceremonies	1	CO2
	Product Vision, Roadmap, Release and Sprint Planning	1	CO2, CO6
Module 3	Daily Scrum and Measuring Progress in Agile Projects	1	CO3, CO6
	Burndown Charts, Velocity and Cycle Time	1	CO3, CO6
Module 4	Introduction to Agile Project Management Tools	1	CO4, CO6
	ZOHO Sprints, GitHub, Jira, Trello and Asana	2	CO4, CO6
	Tool Selection, Agile Tool Integration and Version Control	1	CO4
	Creating a Project in ZOHO Sprints	1	CO4, CO6
Module 5	Agile Team Dynamics, Building High-Performing Agile Teams	1	CO5, CO6
	Communication in Agile Environments, Distributed Agile Teams and Challenges	1	CO5, CO6
Module 6	Challenges in Scaling Agile, SAFe, LeSS and Nexus Frameworks	1	CO5
	Agile at Enterprise Level, Customizing Agile Practices	1	CO5, CO6
	Agile Transformation in Startups, Role of AI in Agile and Agile Adoption in Corporate Environment	1	CO6
	Total Periods	**24**	

List of Experiments with CO Mapping

Sl. No.	Name of the Experiment	Level of Experiment	Objective(s)	COs Addressed
1	Write Agile user stories and acceptance criteria from given software requirements	3	Design	CO3

2	Explore Zoho Sprints tool for Scrum project management framework and Github	3	Experimentation	CO2, CO6
3	Estimate effort for user stories using Planning Poker and story points.	3	Data Analysis	CO3, C06
4	Create and prioritize a product backlog using Agile prioritization techniques.	3	Design	CO3
5	Perform sprint planning activity by selecting backlog items and defining sprint goals.	3	Design	CO2
6	Design and manage a Scrum board to track task progress during development	3	Data Analysis	CO4
7	Prepare and analyze a Sprint Burndown Chart to monitor sprint progress.	3	Data Analysis	CO4, CO6
8	Implement a Kanban workflow with work-in-progress limits and task tracking using any tool	3	Experimentation	CO5, CO6

Course Designer(s):

1. Dr. N. Shivakumar, Associate Professor, CSE, shiva@tce.edu
2. Mrs.S.Saradha, Assistant Professor, CSE, ssacse@tce.edu

26PDCE0	ALGORITHM DESIGN AND ANALYSIS LAB	Category	L	T	P	Credits
		PC	0	0	3	1.5

Preamble

This course provides practical experience in implementing and analyzing algorithms for solving computational problems. The laboratory covers graph, cryptographic, parallel, heuristic, and randomized algorithms through hands-on experiments and research-oriented activities.

Prerequisite

- Programming in C/C++/Java
- Data Structures and Algorithms

Course Outcomes

On the successful completion of the course, students will be able to

COs	Course Outcome 1 (CO1)	TCE Proficiency Scale	SDG Addressed	SDG Level
CO1	Apply dynamic programming strategy to solve optimization problems.	TPS 3	SDG 4	2
CO2	Implement greedy algorithms to solve computational problems.	TPS 3	SDG 9	2
CO3	Apply graph algorithms to solve computational problems.	TPS 3	SDG 9	2
CO4	Illustrate the network flow problem for combinatorial problems.	TPS 3	SDG 9	2
CO5	Analyze the various approximation algorithms with proven ratios and apply fixed-parameter tractable algorithms and kernelization for parameterized problems.	TPS 4	SDG 4	2
CO6	Analyze the different randomized algorithm techniques with rigorous probabilistic analysis	TPS 4	SDG 4	2
CO7	Analyze research-oriented algorithmic techniques for solving complex optimization problems using appropriate computational approaches.	TPS 4	SDG 9	2

Mapping with Programme Outcomes

COs	PO 1	PO 2	PO3	PO 4	PO 5	PO6	PO7	PO8	PO9	PO10	PO 11
CO1.	M	M	M	L	M			S	M	L	L
CO2.	M	M	M	L	M			S	M	L	L
CO3.	M	M	M	L	M			S	M	L	L
CO4.	M	M	M	L	M			S	M	L	L
CO5.	S	S	S	S	M			S	M	L	L
CO6.	S	S	S	S	M	M		S	M	L	L
CO7.	S	S	S	S	M	M	M	S	M	L	L

Assessment Pattern: Cognitive Domain

Type of Assessment	Component	Marks	Remarks
Continuous Assessment	Pre-lab, In-lab, Post-lab	75	Conducted throughout the semester
Model Test	CO-wise (Cognitive, Psychomotor, Affective)	25	Mid/End semester internal test
Total Internal Marks		**100**	
End Semester Examination	CO-wise (Cognitive, Psychomotor, Affective)	100	As directed by COE or as per curriculum

List of Experiments with CO Mapping

Sl. No.	Name of the Experiment	Level of Experiment	Objective(s)	COs Addressed
1.	LCS & Edit Distance using Dynamic Programming	Level 1	Data Analysis	CO1

2.	Needleman-Wunsch & Suffix Arrays using Dynamic Programming	Level 1	Experimentation	CO1
3.	Weighted Job Scheduling using Greedy	Level 1	Experimentation	CO2
4.	Kosaraju's SCC	Level 2	Creativity	CO3
5.	Floyd-Warshall All-Pairs Shortest Path	Level 1	Instrumentation, Design	CO3
6.	Ford-Fulkerson & Edmonds-Karp	Level 2	Design	CO4
7.	Vertex Cover using Approximation Algorithm	Level 2	Data Analysis	CO5
8.	Randomized Quick Sort	Level 2	Design	CO6
9.	Karger’s Min-Cut algorithm	Level 3	Creativity	CO6
10.	Design and implement a collaborative mini-project for solving a real-world optimization problem using suitable algorithmic approaches and present the findings as a technical report	Level 3	Teamwork, Communication, Ethics & Integrity	CO7

Course Designer(s):

1. Raja Lavanya, Assistant Professor, Department of Computer Science and Engineering — rlit@tce.edu

2. Dharani J, Assistant Professor, Department of Computer Science and Engineering — jdicse@tce.edu

CURRICULUM AND DETAILED SYLLABI

FOR

M.E DEGREE (Computer Science and Engineering) PROGRAMME

PROGRAMME ELECTIVE

FOR THE STUDENTS ADMITTED FROM THE

ACADEMIC YEAR 2026 - 2027 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING

(A Government Aided ISO 9001-2008 certified

Autonomous Institution affiliated to Anna University)

MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41

Fax: 0452 2483427

Web: www.tce.edu

		L	T	P	Credit
26PDPA0	**DATA MODELLING AND VISUALIZATION**	3	0	0	3

1. Preamble

The rapid growth of data in engineering and business domains demands professionals skilled in data organization, analysis, visualization, and processing. This course provides a strong foundation in data modelling, statistical and exploratory analysis, data visualization, dashboard design, and real-time data processing using relational, ER, and NoSQL approaches. It prepares students for advanced Machine Learning studies by strengthening core skills in data representation, statistical reasoning, and visual analytics, while aligning with current industry practices.

2. Prerequisite

- Database Management Systems
- Basic Python / R Programming
- Fundamentals of Probability and Statistics

3. Course Outcomes

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

No.	Course Outcome	TPS Level	CO Weightage	SDG addressed	SDG Level
CO1	Apply data modelling techniques to design relational, dimensional, and semi-structured schemas for data management requirements.	TPS3	10%	SDG 9	1
CO2	Use exploratory data analysis techniques, statistical computing libraries, and data preprocessing methods to prepare and analyze datasets for decision-making.	TPS3	20%	-	-
C03	Apply statistical modelling, hypothesis testing, probability concepts, and linear algebra techniques to solve data analytics problems and interpret analytical results.	TPS3	20%	-	-
CO4	Implement NoSQL database solutions for scalable and distributed data management.	TPS3	20%	SDG 9	1
CO5	Develop visual analytics and interactive dashboards using appropriate visualization techniques and tools to communicate insights effectively.	TPS3	20%	-	-
CO6	Use stream processing, time-series forecasting, and real-time analytics methods to support monitoring and operational decision-making.	TPS3	10%	SDG 11	1

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	S	M	S	L	L	L			L		L
CO2	M	S	S	M	S			L	M		L
CO3	S	S	S	M	M			L	M		L
CO4	M	M	S	L	S	L			L		L
CO5	M	M	S	L	S	L		S	L		L
CO6	M	S	S	M	S	L		L	M		L

4. Assessment Pattern

CO	CAT1	Assignment 1	CAT2	Assignment 2	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	30	30			10
CO2	40	40			20
CO3	30	30			20
CO4			30	30	20
CO5			40	40	20
CO6			30	30	10
Total	100	100	100	100	100

5. Syllabus

Module 1: Data Modelling — Representation, Structure and Schema Design

Introduction to data systems and data-driven computing — Types of data: structured, semi-structured, and unstructured — Operational and analytical data systems: OLTP and OLAP concepts - Data representation concepts: conceptual, logical, and physical models; schema design workflow — Relational data model and ER modelling: entities, attributes, relationships, cardinality, participation constraints, and Extended ER concepts — Mapping ER diagrams to relational schemas — Dimensional modelling: facts, dimensions, measures, hierarchies, Star schema, and Snowflake schema — Semi-structured data models: XML (DTD, XSD) and JSON Schema — UML Class Diagrams: notation, associations, and aggregation — Overview of modern data storage paradigms.

Module 2: Exploratory Data Analysis

Data science lifecycle — business understanding, data understanding, data preparation, modelling, evaluation, and deployment — Motivation and objectives of Exploratory Data Analysis (EDA) — Data preprocessing and cleaning: handling missing values, duplicates, and inconsistencies — Data transformation and normalization: min-max scaling and z-score standardization — Exploratory statistical analysis: measures of central tendency, dispersion, distribution characteristics, and data skewness — Pattern discovery and anomaly detection — Statistical computing libraries and related concepts: NumPy, SciPy, Pandas, and basic statistical functions for data analysis — EDA workflows using Python (Pandas, NumPy, Matplotlib): examples on real datasets.

Module 3: Statistical Modelling and Analytical Reasoning

Review of probability and statistics for data analytics — Descriptive statistics: mean, median, mode, variance, and standard deviation — Probability fundamentals: conditional probability and Bayes' theorem — Common probability distributions: Binomial and Normal distributions — Correlation and covariance — Sampling and estimation: point estimation and sampling distributions — Statistical inference: confidence intervals and hypothesis testing — Hypothesis testing concepts: null and alternative hypotheses, p-value, significance level, Type I and Type II errors — t-test and chi-square test — Linear algebra foundations for data analytics: vectors, matrices, matrix operations, systems of linear equations, eigenvalues and eigenvectors — Applications of linear algebra in data analysis and dimensionality reduction — Introduction to predictive modelling: simple linear regression, model fit metrics (R^2 and RMSE), and residual analysis.

Module 4: NoSQL and Scalable Data Models

Limitations of relational databases for big data workloads — Motivation for NoSQL systems — CAP theorem and BASE properties — Document-oriented databases: JSON and BSON structures, embedded vs. referenced document design patterns — Key-value stores: principles, use cases - Column-family databases: wide-column storage and query patterns — Graph databases: nodes, edges, properties, and traversal query patterns — Data partitioning, replication, and sharding concepts — Comparative study of SQL and NoSQL systems; selection criteria for different application scenarios.

Module 5: Data Visualization and Predictive Modelling Dashboards

Principles of effective data visualization — pre-attentive attributes, Gestalt principles, colour theory, and visual hierarchy — Core chart types: histograms, scatter plots, box plots, heat maps, and time-series plots — Advanced visualizations: treemaps, network/graph visualizations, knowledge graphs for entity relationship visualization, ontology graphs for semantic data representation, and geo-spatial maps (choropleth and bubble maps).— Visualization tools: Matplotlib, Seaborn, Plotly (Python), Power BI, and Tableau overview — Predictive modelling visualization: regression curves, residual plots, ROC curves, confusion matrix heatmaps, and feature importance charts — Interactive dashboards and KPI panels: design principles, drill-down, and filtering — Storytelling with data: narrative design and insight communication; case studies in healthcare, finance, and retail analytics.

Module 6: Stream Processing and Real-Time Analytics

Real-time vs. batch analytics — architecture comparison, streaming data characteristics, and use cases —Apache Spark Structured Streaming: DStream vs. DataFrame API, micro-batch and continuous processing modes — Time series analysis: components (trend, seasonality, cyclical, irregular), ARIMA model, exponential smoothing, and forecasting — Real-time dashboards and event-driven analytics —Applied scenarios: IoT sensor streams, social media feeds, financial tick data.

6. Learning Resources
6.1 Text Books

1. Dr. Gypsy Nandi, Dr. Rupam Kumar Sharma, Data Science Fundamentals and Practical Approaches, BPB Publications, 2020.
2. Wes McKinney, "Python for Data Analysis", 3rd Edition, O'Reilly Media, 2022
3. Cole Nussbaumer Knaflic, "Storytelling with Data: A Data Visualization Guide for Business Professionals", Wiley, 2015.
4. Peter Bruce, Andrew Bruce, Practical Statistics for Data Scientists, O'Reilly Publishers, 2018.
5. Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee, "Learning Spark", 2nd Edition, O'Reilly Media, 2020.

6.2 Reference Books & Web resources

1. RamezElmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", 7th Edition, Pearson Education, 2017.
2. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.
3. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, "An Introduction to Statistical Learning", 2nd Edition, Springer, 2021
4. Douglas C. Montgomery and George C. Runger, "Applied Statistics and Probability for Engineers", 7th Edition, Wiley, 2022.
5. NPTEL – Data Science for Engineers (IIT Madras): https://nptel.ac.in
6. Kaggle Learn – EDA, Data Visualization: https://www.kaggle.com/learn

7. Course Contents and Lecture Schedule

Module No	Topics	No of Periods
Module 1: Data Modelling — Representation, Structure and Schema (5 Hours)		
1.1	Introduction: data systems, data-driven computing, and types of data (structured, semi-structured, unstructured), Operational and analytical data systems: OLTP and OLAP concepts	1
1.2	Data representation concepts — conceptual, logical, and physical models; schema design workflow	1
1.3	Relational data model and ER modelling: entities, attributes, relationships, cardinality, participation constraints, Extended ER concepts — Mapping ER diagrams to relational schemas	1
1.4	Dimensional modelling: facts, dimensions, measures, hierarchies, Star schema, and Snowflake schema	1
1.5	Semi-structured data models: XML (DTD, XSD) and JSON Schema	1

Module No	Topics	No of Periods
1.6	UML Class Diagrams: notation, associations, and aggregation — Overview of modern data storage paradigms	
Module 2: Exploratory Data Analysis (7 Hours)		
2.1	Data science lifecycle — business understanding, data understanding, data preparation, modelling, evaluation, deployment	1
2.2	Motivation and objectives of Exploratory Data Analysis (EDA) - Data preprocessing and cleaning — handling missing values, duplicates, inconsistencies	1
2.3	Data transformation and normalization — min-max scaling, z-score standardization	1
2.4	Exploratory statistical analysis: measures of central tendency, dispersion, distribution characteristics, and data skewness	1
2.5	Pattern discovery and anomaly detection in datasets - Statistical computing libraries and related concepts: NumPy, SciPy, Pandas, and basic statistical functions for data analysis	2
2.6	EDA workflows using Python (Pandas, NumPy, Matplotlib) — Examples on real datasets.	1
Module 3: Statistical Modelling and Analytical Reasoning (7 Hours)		
3.1	Review of probability and statistics for data analytics - Descriptive statistics — mean, median, mode, variance, standard deviation	1
3.2	Probability fundamentals: conditional probability and Bayes' theorem — Common probability distributions: Binomial and Normal distributions	1
3.3	Correlation and covariance — Sampling and estimation: point estimation and sampling distributions	1
3.4	Statistical inference: confidence intervals and hypothesis testing — Hypothesis testing concepts: null and alternative hypotheses, p-value, significance level, Type I and Type II errors — t-test and chi-square test	2
3.5	Linear algebra foundations for data analytics: vectors, matrices, matrix operations, systems of linear equations, eigenvalues and eigenvectors, Applications of linear algebra in data analysis and dimensionality reduction	1

Module No	Topics	No of Periods
3.6	Introduction to predictive modelling: simple linear regression, model fit metrics (R^2 and RMSE), and residual analysis.	1
Module 4: NoSQL and Scalable Data Models (6 Hours)		
4.1	Limitations of relational databases for big data workloads - motivation for NoSQL systems. CAP theorem and BASE properties	1
4.2	Document-oriented databases — JSON/BSON structures, embedded vs. referenced documents	1
4.3	Key-value stores: principles, use cases - Column-family databases: wide-column storage and query patterns	2
4.4	Graph databases — nodes, edges, properties, and traversal query patterns	1
4.5	Data partitioning, replication, and sharding concepts, Comparative study — SQL vs. NoSQL; selection criteria for application scenarios	1
Module 5: Data Visualization and Predictive Modelling Dashboards (6 Hours)		
5.1	Principles of effective data visualization — pre-attentive attributes, Gestalt principles, colour theory, visual hierarchy	1
5.2	Core chart types — histograms, scatter plots, box plots, heat maps, time-series; chart selection framework	1
5.3	Advanced and multidimensional visualizations — treemaps, network/ graph visualizations, knowledge graphs for entity relationship visualization, ontology graphs for semantic data representation, geo-spatial maps	1
5.4	Visualization tools — Matplotlib, Seaborn, Plotly (Python), Power BI and Tableau overview	1
5.5	Predictive modelling visualization — regression curves, residual plots, ROC curves, confusion matrix, heatmaps and feature importance charts	1
5.6	Interactive dashboards and KPI panels — design principles, drill-down, filtering	1
5.7	Storytelling with data — narrative design, insight communication; case studies (healthcare, finance, retail)	

Module No	Topics	No of Periods
Module 6: Stream Processing and Real-Time Analytics (5 Hours)		
6.1	Real-time vs. batch analytics — architecture comparison, and streaming data characteristics and use cases	1
6.2	Apache Spark Structured Streaming — DStream vs. DataFrame API, micro-batch and continuous processing modes	2
6.3	Time series analysis — components (trend, seasonality, cyclical, irregular), ARIMA model, exponential smoothing, forecasting	1
6.4	Real-time dashboards and event-driven analytics — Applied scenarios: IoT sensor streams, social media feeds, financial tick data.	1
	Total Periods	**36**

8. Course Designer(s):

1. Dr.B.Subbulakshmi, Professor, bscse@tce.edu,CSE department
2. D.Priyadharshini , Assistant Professor, dpicse@tce.edu , CSE department

M.E. COMPUTER SCIENCE AND ENGINEERING DEGREE PROGRAMME

CURRICULUM AND DETAILED SYLLABI

FOR

SECOND SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2026-27 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
M.E. Degree Programme

COURSES OF STUDY
(For the candidates admitted from 2026-27 onwards)

SECOND SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
26PDCF0	Cloud Computing Systems and Services	PCC	3	-	-	3
26PDPX0	Programme Elective - 2	PEC	3	-	-	3
26PDPX0	Programme Elective - 3	PEC	3	-	-	3
	Research Methodology and IPR		3	-	-	3
THEORY CUM PRACTICAL						
26PDCG0	Deep Learning	PCC	2	-	2	3
PRACTICAL						
26PDCH0	Systems Programming Lab	PCC	-	-	4	2
PROJECT						
	Mini Project	Project				3
		Total				**20**

PCC : Programme Core Courses
PEC : Programme Elective Courses

L : Lecture
T : Tutorial
P : Practical

Note:
1 Lecture Period/week is equivalent to 1 credit
1 Tutorial Period/week is equivalent to 0.5 credit
1 Practical Period/week is equivalent to 0.5 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
M.E Degree Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2026-27 onwards)

SECOND SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	26PDCF0	Cloud Computing Systems and Services	3	40	60	100	27	50
2	26PDPX0	Programme Elective - 2	3	40	60	100	27	50
3	26PDPX0	Programme Elective - 3	3	40	60	100	27	50
4		Research Methodology and IPR	3	40	60	100	27	50
THEORY CUM PRACTICAL								
5	26PDCG0	Deep Learning	3 (Terminal Exam: Theory)	50	50	100	22.5	50
PRACTICAL								
6	26PDCH0	Systems Programming Lab	3	60	40	100	18	50
7		Mini Project	3	60	40	100	18	50

26PDCF0	**CLOUD COMPUTING SYSTEMS AND SERVICES**	Category	L	T	P	Credit
		PCC	3	0	0	3

Preamble

This course builds foundational knowledge of cloud architectures, virtualization, and service models. It equips students to design, deploy, and manage secure cloud-native applications using modern distributed platforms, microservices, and AI-driven orchestration across edge, serverless, and federated environments.

Prerequisite

NIL

Course Outcomes

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

No.	Course Outcome	TPS Level	CO Weight age	SDG address ed	SDG Level
CO1	Use evolution of computing to explain business drivers for cloud adoption.	TPS3	15	-	-
CO2	Solve multi-container applications using Docker and Kubernetes with exposed services.	TPS3	15	-	-
CO3	Compute load balancing SLA policies monitoring and AI-driven auto scaling to optimize performance	TPS3	20	-	-
CO4	use REST APIs API gateways microservices and serverless functions to build and orchestrate scalable cloud services	TPS3	15	-	-
CO5	Apply security techniques IAM encryption and monitoring to protect cloud services and ensure system reliability.	TPS3	15	-	-
CO6	Analyze emerging cloud paradigms and intelligent systems to evaluate their impact on scalability, cost, and sustainability.	TPS4	20	9,11	2

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	S	M				S			M		L
CO2	S	S	S		S	M					L
CO3	S	S	S	M	S	M		L		M	L
CO4	S	M	S		S	M		M			L
CO5	S	S	S	S	S	S	S				
CO6	S	S	S	S	S	S				M	S

Assessment Pattern

CO	CAT1	Assignment 1	CAT2	Assignment 2	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	30	30			15
CO2	30	30			15
CO3	40	40			20
CO4			30	30	15
CO5			30	30	15
CO6			40	40	20
Total	**100**	**100**	**100**	**100**	**100**

Syllabus

Fundamentals of Cloud Computing and Architectures - Evolution - Cloud Service Models - Cloud Deployment Models - Cloud Reference Architecture and Data Center Design -Virtualized Data Center and Software Defined Data Center (SDDC) - Hyperscale Cloud Systems.

Virtualization and Cloud Infrastructure Technologies - Virtualization and Hypervisors- Hypervisor Energy Management - Virtual Machines and Resource Provisioning - Containerization: Docker and Kubernetes - Infrastructure as Code: Terraform and Ansible - Scheduling Algorithms - Elastic Scaling and Load Balancing.

Cloud Services and Service Management - Cloud Service Architecture - RESTful and SOAP based Services -API Gateway and API Management - Microservices and Event-driven Architecture - Serverless Computing (FaaS) - Service Orchestration- SLA and QoS Management- Monitoring and Observability - Multi-cloud Integration- AI-driven Service Optimization.

Cloud Security, Reliability and Observability-Cloud Security Architecture - IAM, OAuth2 and JWT- Secure APIs and Container Security- Encryption Techniques- Disaster Recovery-Site Reliability Engineering- Monitoring and Distributed Tracing - DevSecOps and AI Security- Cloud Security Posture Management.

Emerging Cloud Paradigms and Intelligent Cloud Systems - Next-Gen Cloud Models: Serverless & FaaS 2.0, Edge/Cloud Continuum, Intelligent Infrastructure: ML-driven Auto-scaling, Predictive Maintenance, Cloud-Native AI/ML, Sustainable & Specialized Clouds: Green Cloud Computing.

6. Learning Resources

6.1 Text Book

1. P Brendan Burns, Joe Beda and Kelsey Hightower, Kubernetes: Up and Running, 3rd Edition, O’Reilly Media, 2022.
2. Sam Newman, Building Microservices: Designing Fine-Grained Systems, 2nd Edition, O’Reilly Media, 2021.
3. Justin Domingus and John Arundel, Cloud Native DevOps with Kubernetes, O’Reilly Media, 2022.
4. Marko Lukša and Kevin Conner, Kubernetes in Action, 2nd Edition, Manning Publications, 2025.

6.2 Reference Books & Web resources

1. Dan C. Marinescu, Cloud Computing: Theory and Practice, Morgan Kaufmann Publishers, 2nd Edition, 2017.
2. Michael J. Kavis, Architecting the Cloud: Design Decisions for Cloud Computing Service Models, Wiley Publications, 2014.
3. Lee Calcote and Zak Butcher, Istio: Up and Running – Using a Service Mesh to Connect, Secure, Control, and Observe, O'Reilly Media, 2022.
4. Justin Garrison and Kris Nova, Cloud Native Infrastructure, O'Reilly Media, 2017.
5. Evi Nemeth, Garth Snyder, Trent R. Hein and Ben Whaley, UNIX and Linux System Administration Handbook, Pearson Education, 5th Edition, 2017.

7. Course Contents and Lecture Schedule

Module no.	Topic	No. of lectures
Fundamentals of Cloud Computing and Architectures (6)		
1.1	Evolution	2
1.2	Cloud Service Models	
1.3	Cloud Deployment Models	
1.4	Cloud Reference Architecture and Data Center Design	1
1.5	Virtualized Data Center and Software Defined Data Center (SDDC)	2
1.6	Hyperscale Cloud Systems	1
Virtualization and Cloud Infrastructure Technologies (9)		
2.1	Virtualization and Hypervisors	1
2.2	Hypervisor Energy Management	1
2.3	Virtual Machines and Resource Provisioning	2
2.4	Containerization: Docker and Kubernetes	2
2.5	Infrastructure as Code using Terraform and Ansible	2
2.6	Scheduling Algorithms	
2.7	Elastic Scaling and Load Balancing	1
Cloud Services and Service Management (6)		
3.1	Cloud Service Architecture	1
3.2	RESTful and SOAP	

3.3	API Gateway and API Management	1
3.4	Microservices and Event-driven Architecture	
3.5	Serverless Computing (FaaS)	1
3.6	Service Orchestration	
3.7	SLA and QoS Management	1
3.8	Monitoring and Observability	1
3.9	Multi-cloud Integration	1
3.10	AI-driven Service Optimization	
Cloud Security, Reliability and Observability (8)		
4.1	Cloud Security Architecture	2
4.2	IAM, OAuth2 and JWT	
4.3	Secure APIs and Container Security	1
4.4	Encryption Techniques	1
4.5	Disaster Recovery	1
4.6	Site Reliability Engineering	2
4.7	Monitoring and Distributed Tracing	
4.8	DevSecOps and AI Security	1
4.9	Cloud Security Posture Management	
Emerging Cloud Paradigms and Intelligent Cloud Systems (7)		
5.1	Next-Gen Cloud Models	2
5.2	Intelligent Infrastructure	2
5.3	Cloud-Native AI/ML	1
5.4	Sustainable & Specialized Clouds:G reen Cloud Computing	2
	Total Periods	**36**

8. Course Designer(s):

1. Dr.C.Santhiya, Assistant Professor, CSE, csit@tce.edu
2. Dr.K.Indira, Assistant Professor, CSE, kiit@tce.edu

26PDCG0	DEEP LEARNING	Category	L	T	P	Credit	TE Type
		PCC	2	0	2	3	TCP-T

Preamble

This Theory cum Practical (TCP) course equips postgraduate students in Computer Science and Engineering with a rigorous and hands-on foundation in deep learning. This course helps build a foundation in deep learning and its variants such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), and explores how they can be applied to solve problems in domains like computer vision, speech processing, and natural language processing (NLP).

Prerequisite

NIL

Course Outcomes

- On the successful completion of the course, students will be able to

No.	Course Outcome	TCE Proficiency Scale	POs addressed	CO Weightage	SDG Addressed	SDG Level
CO1	Analyze the evolution, paradigms, and comparative relationships among Artificial Intelligence, Machine Learning, and Deep Learning and their real-world applications.	TPS 2	PO1, PO2, PO3, PO4, PO9	10	SDG 9: Industry, Innovation & Infrastructure	3
CO2	Design and train neural networks by applying mathematical foundations, activation functions, loss functions, gradient descent, backpropogation and weight initialization	TPS 3	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11	18	SDG 4: Quality Education	2
CO3	Develop deep neural networks using regularization, representation learning, auto encoders and frameworks	TPS 3	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11	18	SDG 9: Industry, Innovation & Infrastructure	3

CO4	Evaluate CNNs, RNNs, LSTM, GRU, and attention mechanisms for image, sequence and text tasks	TPS 3	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11	18	SDG 4: Quality Education	3
CO5	Investigate Transformers, BERT, GPT, GANs, VAEs and Deep Reinforcement Learning for real-world applications.	TPS 3	PO1, PO2, PO3, PO4, PO5, PO6, PO8, PO9, PO10, PO11	18	SDG 10: Reduced Inequalities	2
CO6	*Evaluate emerging research in Explainable AI, Federated Learning, Edge AI, Computer Vision, NLP, Cyber Security and Healthcare Analytics.*	TPS 4	PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11	18	SDG 16: Peace, Justice & Strong Institutions	4

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	S	M	L	L					L		
CO2	S	S	M	L	M	L		L	L	L	M
CO3	S	M	L	L	M	L		L	L	L	M
CO4	S	M	L	L	M	L		L	L	L	M
CO5	S	M	L	L	M	L		L	L	L	M
CO6	S	S	M	L	M	L	L	L	L	L	M

S- Strong; M-Medium; L-Low

Assessment Pattern

CO	CAT1	CAT2	Model Test CO-wise (Cognitive, Psychomotor, Affective)	OCR/Continuous Assessment (Pre-lab, In-lab, Post-lab)	Terminal Exam
	% of Marks	% of Marks	% of Marks	% of Marks	% of Marks
CO1	20		10	10	10
CO2	40		18	18	18
CO3	40		18	18	18
CO4		30	18	18	18
CO5		40	18	18	18

CO6		30	18	18	18
Total	100	100	100	100	100

Assessment Pattern: Practical Component

Type	Component	Marks
Continuous Assessment	Pre lab In lab Post lab	75
Model Test	CO wise practical assessment	25
Total Internal		100
End Semester Exam	Theory Examination	100

Syllabus

Module I Foundations of Artificial Intelligence, Machine Learning and Deep Learning

Evolution of Artificial Intelligence - Knowledge Representation and Intelligent Systems -Machine Learning Paradigms -Supervised Learning -Unsupervised Learning -Reinforcement Learning -Limitations of Conventional Machine Learning -Deep Learning Fundamentals -AI, ML, and DL: Comparative Analysis -Deep Learning Software Ecosystem and applications

Module II Mathematical Foundations and Neural Networks

Probability and Statistics Fundamentals -Optimization Concepts -Artificial Neuron Model-Perceptron and Multilayer Perceptron -Activation Functions -Feed Forward Neural Networks-Error Functions and Loss Functions -Gradient Descent Optimization -Backpropagation Algorithm -Weight Initialization Strategies

Module III Deep Neural Networks and Representation Learning

Deep Feed Forward Networks -Universal Approximation Theorem -Vanishing and Exploding Gradient Problems -Batch Normalization -Dropout and Regularization Techniques -Hyperparameter Optimization -Representation Learning –Autoencoders -Sparse and Denoising Autoencoders -Deep Learning Frameworks –TensorFlow –PyTorch -Keras

Module IV Deep Learning Architectures

Convolutional Neural Networks-Convolution Operations -Pooling Methods -CNN Architectures –LeNet –AlexNet –VGGNet –ResNet -Transfer Learning -Recurrent Neural Networks-Sequence Modeling -RNN Architecture -Long Short-Term Memory (LSTM) -Gated Recurrent Units (GRU) -Attention Mechanisms

Module V Advanced Deep Learning

Transformer Architecture -BERT Model and GPT Model -Generative Adversarial Networks (GANs) –Variational Autoencoders (VAEs) -Deep Reinforcement Learning – identifying relevant application domains

Module VI Research Applications

Explainable Artificial Intelligence (XAI) -Federated Learning -Edge AI and TinyML -Deep Learning for -Computer Vision -Natural Language Processing -Cyber Security -Healthcare Analytics -Emerging Research Trends in Deep Learning

Text Book

1. Goodfellow, I., Bengio, Y., &Courville, A. *Deep learning*. MIT Press, 2016.
2. Aggarwal, Charu C. *Neural Networks and Deep Learning*. Springer, 2018.
3. Chollet, Francois. *Deep Learning with Python*. Manning Publications, 2017.

Reference Books& web resources

1. Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.
2. Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
3. Russell, Stuart J., and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th ed., Pearson, 2020.
4. Géron, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media, 2019.

SDG Alignment

Activities aligned with SDGs addressed:

1. SDG 4 Quality Education: Students develop programs for basic learning support tasks such as quiz evaluation, simple grading utilities, and self practice problem solvers to understand how computing can improve access to quality learning tools.
2. SDG 9 Industry Innovation and Infrastructure: Students design applications using deep learning to solve real world problems related to digital infrastructure and software innovation

Course Contents and Lecture Schedule

Module No	Topics	No of Periods	**Course Outcomes**
1	**Foundations of Artificial Intelligence, Machine Learning and Deep Learning**		
1.1	Evolution of Artificial Intelligence -Knowledge Representation and Intelligent Systems	1	CO1
1.2	Machine Learning Paradigms	1	CO1
1.3	AI, ML, and DL: Comparative Analysis	1	CO1
1.4	Deep Learning Software Ecosystem and applications	1	CO1
2	**Mathematical Foundations and Neural Networks**		
2.1	Probability and Statistics Fundamentals - Error Functions and Loss Functions	1	CO2
2.2	Gradient Descent Optimization	1	CO2
2.3	Backpropagation Algorithm	1	CO2
2.4	Weight Initialization Strategies	1	CO2
3	**Deep Neural Networks and Representation Learning**		
3.1	Deep Feed Forward Networks - Universal Approximation Theorem	1	CO3
3.2	Representation Learning - Autoencoders	1	CO3
3.3	Sparse and Denoising Autoencoders	1	CO3
3.4	Deep Learning Frameworks- TensorFlow - PyTorch–Keras	1	CO3
4	**Deep Learning Architectures**		
4.1	Convolutional Neural Networks - CNN Architectures	1	CO4
4.2	Transfer Learning	1	CO4
4.3	Recurrent Neural Networks	1	CO4
4.4	Sequence Modeling - RNN Architecture - Long Short-Term Memory (LSTM)	1	CO4
5	**Advanced Deep Learning**		
5.1	Transformer Architecture	1	CO5
5.2	BERT and GPT Models	1	CO5
5.3	Generative Adversarial Networks (GANs) - Variational Autoencoders (VAEs)	1	CO5

5.4	Deep Reinforcement Learning – identifying relevant application domains	1	CO5
6	**Research Applications**		
6.1	Explainable Artificial Intelligence (XAI)	1	CO6
6.2	Federated Learning - Edge AI and TinyML	1	CO6
6.3	Deep Learning for Computer Vision, NLP, Cyber Security, Health care analytics	1	CO6
6.4	Emerging Research Trends in Deep Learning	1	CO6
	Total Hours	**24**	

List of Experiments with CO Mapping

Sl. No.	Name of the Experiment	Level of Experiment	COs Addressed	Objective(s)
1	Implementation of Neural Networks from Scratch.	1	CO2	Experimentation, Data Analysis, Models
2	Image Classification using CNNs.	1	CO4	Models, Data Analysis, Creativity
3	Transfer Learning using Pre-trained Models.	1	CO4	Instrumentation, Models, Communication
4	Text Classification using LSTM and GRU.	2	CO4	Experimentation, Data Analysis, Design
5	Transformer-based Language Models	2	CO4	Design, Learn from Failure, Data Analysis
6	Autoencoder-based Feature Learning.	2	CO3	Models, Experimentation, Ethics & Integrity
7	GAN-based Synthetic Data Generation.	3	CO5	Instrumentation, Sensory Awareness, Ethics & Integrity
8	Integrated mini project	3	CO5, CO6	Experimentation, Data Analysis, Models

Course Designer(s):

Dr. K. Sundarakantham, Professor(CAS), CSE	kskcse@tce.edu
Dr. S. Mercy Shalinie, Professor and Head , CSE	shalinie@tce.edu

26PDCH0	SYSTEMS PROGRAMMING LAB	Category	L	T	P	Credits
		PCC	0	0	3	1.5

Preamble

This laboratory course provides hands-on exposure to systems programming concepts through the development of programs involving operating system utilities, data structures, process and thread management, inter-process communication, socket programming, server-side scripting, and code optimization. The course enables students to design, implement, analyze, and evaluate system-level applications using contemporary programming tools and practices.

Prerequisite

Programming in C/Java/Python, basic data structures, operating system fundamentals, and introductory computer networks.

Course Outcomes

On the successful completion of the course, students will be able to

COs	Course Outcome	TCE Proficiency Scale	SDG Addressed	SDG Level
CO1	Apply Linux/POSIX shell scripting, process utilities, pipes, filters, and resource-monitoring commands to automate system-level tasks.	TPS 3	SDG 9	1
CO2	Apply abstract data types, searching, sorting, and graph-traversal algorithms to develop efficient systems programming utilities.	TPS 3	SDG 9	1
CO3	Implement computational solutions for algorithmic and graph-based problems and validate their correctness using contemporary programming and debugging tools.	TPS 3	SDG 9	1
CO4	Apply process, thread, inter-process communication, synchronization, semaphore, and deadlock-handling mechanisms to solve concurrent systems problems.	TPS 4	SDG 9	1
CO5	Apply TCP/UDP socket programming, client-server design, server-side scripting, and database connectivity to implement networked systems applications.	TPS 4	SDG 9	1
CO6	Apply control-flow analysis, data-flow analysis, profiling, and basic code-optimization techniques to	TPS 3	SDG 12	1

	improve program performance and resource efficiency.			

Mapping with Programme Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	M	L	M	L	M			L			
CO2	M	L	S	L	L						
CO3	M	M	S	M	M			L			L
CO4	M	L	S	M	L						
CO5	M	L	S	M	M	L		M			
CO6	M	M	M	S	S			L		M	L

Assessment Pattern: Cognitive Domain

Type of Assessment	Component	Marks	Remarks
Continuous Assessment	Pre-lab, In-lab, Post-lab	75	Conducted throughout the semester
Model Test	CO-wise (Cognitive, Psychomotor, Affective)	25	Mid/End semester internal test
Total Internal Marks		**100**	
End Semester Examination	CO-wise (Cognitive, Psychomotor, Affective)	100	As directed by COE or as per curriculum

List of Experiments with CO Mapping

Sl. No.	Name of the Experiment	Level of Experiment	Objective(s)	COs Addressed
1	Linux/POSIX shell scripting for process attributes and resource monitoring	1	Experimentation; Data Analysis; Communication	CO1
2	Shell automation for file, permission, log, and backup management	1	Design; Learn from Failure; Communication	CO1
3	Implementation of stack and queue applications	1	Design; Psychomotor Skills; O11 Communication	CO2
4	Implementation and empirical analysis of searching and sorting algorithms	2	Experimentation; Data Analysis; Learn from Failure	CO2, CO3
5	Implementation of graph representation and traversal algorithms	2	Models; Design; Data Analysis	CO2, CO3
6	Threading applications and synchronization primitives	2	Experimentation; Design; Learn from Failure	CO4
7	Semaphores and deadlock detection/avoidance	2	Models; Experimentation; Data Analysis	CO4

8	Inter-process communication using pipes, FIFO, shared memory, or message queues	2	Experimentation; Design; Data Analysis	CO4
9	Stream and datagram socket programming	2	Design; Experimentation; Communication	CO5
10	Server-side scripting with database connectivity	2	Design; Safety; Ethics & Integrity; Communication	CO5
11	Control-flow and data-flow analysis	2	Models; Data Analysis; Communication	CO6
12	Simple code optimization and profiling	3	Experimentation; Data Analysis; Learn from Failure; Creativity	CO6

Course Designer(s):

S.No	Name & Designation	Mail ID
1.	Dr. M. Sivakumar, Associate Professor, Dept. of CSE, TCE	mskcse@tce.edu
2.	Mr. S. Santhana Hari, Assistant Professor, Dept. of CSE, TCE	sshcse@tce.edu