SAVITRIBAI PHULE PUNE UNIVERSITY

(Formerly University of Pune)

Faculty of Science and Technology

SYLLABUS BOOKLET

T. E. (THIRD YEAR) COMPUTER ENGINEERING

2024 PATTERN

NATIONAL EDUCATION POLICY (NEP) – 2020

Compliant Curriculum

With Effect from

Academic Year 2026-27



KNOWLEDGE | INNOVATION | SUSTAINABLE DEVELOPMENT

Contents

Abbreviation	**iii**
Preface by Board of Studies	**1**
Program Educational Objectives	**2**
Knowledge and Attitude Profile (WK)	**3**
Program Outcomes	**4**
General Rules	**6**
Curriculum Structure - Semester V	**9**
Curriculum Structure - Semester VI	**10**
Artificial Intelligence	**12**
Computer Networks	**15**
Theory of Computation	**18**
Artificial Intelligence Laboratory	**21**
Computer Networks Lab	**23**
Elective I - Data Mining and Data Warehousing	**26**
Elective I - Cloud Computing	**29**
Elective I - Mobile Computing	**32**
Elective I - Embedded systems	**35**
Elective I - Laboratory	**37**
Robotics and Automation	**42**
Open Elective	**45**
Technical Seminar	**46**
Semester VI - Courses	**50**
Machine Learning	**51**

Software Engineering and Modelling **54**

Machine Learning Lab **57**

Software Engineering and Modelling Lab **60**

Elective II - Quantum Computing **63**

Elective II - Distributed Systems **66**

Elective II - UX/UI Design **68**

Elective II - System Programming **70**

Elective III - Information Retrieval **73**

Elective III - Data Visualization and Analytics **75**

Elective III - Cryptography and Network Security **78**

Elective III - Advanced Databases **81**

Elective III Lab **84**

Green Computing **92**

Solar Technology and Maintenance **98**

Acknowledgement **100**

Nomenclature

AICTE All India Council for Technical Education

CCE Comprehensive Continuous Evaluation

CO Course Outcomes

ELC Experiential Learning Course

ESE End-Semester Examination

GAPC Graduate Attributes and Professional Competencies

MDM Multidisciplinary Minor

NEP National Education Policy

OE Open Elective

PCC Programme Core Course

PEC Programme Elective Course

PEO Programme Educational Objectives

PO Programme Outcomes

PSO Program Specific Outcomes

QPD Question Paper Delivery

SPPU Savitribai Phule Pune University

UGC University Grants Commission

VSE Vocational and Skills Enhancement Course

WK Knowledge and Attitude Profile

Preface by Board of Studies - Computer Engineering

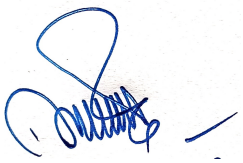
Dear Students and Teachers,

It is a privilege to present the revised curriculum for the T.E. Computer Engineering for the academic year 2026–2027. The field of Computer Engineering is experiencing an unprecedented era of disruption. The rapid evolution and integration of Generative AI, edge computing, advanced cybersecurity architectures, and data-driven systems are fundamentally altering the global industrial landscape. As educators and administrators, our primary responsibility is to ensure that our academic frameworks do not merely react to these transformations, but proactively anticipate them.

Computer Engineering have emerged as transformative forces reshaping industries, driving innovation, and impacting our daily lives. Recognizing the growing importance and pervasive nature of these fields, we have designed this comprehensive syllabus to equip students with the foundational knowledge and practical skills. This curriculum is meticulously crafted to provide a holistic learning experience, blending theoretical concepts with hands-on applications. In framing this syllabus, the Board of Studies (BoS) has focused on a core philosophy: balancing foundational engineering rigor with industry-ready agility. The revised syllabus falls in line with the objectives of NEP-2020, Savitribai Phule Pune University, AICTE, UGC, and various accreditation agencies by keeping an eye on the technological developments, innovations, and industry requirements.

This booklet serves as a comprehensive roadmap for both faculty and students. To our esteemed faculty members, I urge you to look beyond the literal text of the syllabus and bring these subjects to life through innovative, learner-centric pedagogies, active research integration, and industry collaborations. To our students, I encourage you to leverage this curriculum as a launching pad for continuous learning, innovation, and impactful engineering contributions. I extend my deepest gratitude to the members of the Board of Studies, industry advisors, academic peers, and the university leadership for their meticulous inputs, rigorous debates, and unwavering commitment to elevating the standard of engineering education.

Let us collectively strive to nurture a generation of computer engineers who are technically sound, ethically grounded, and fully equipped to lead the global digital frontier.



Dr. Nilesh Uke
Chairman - Board of Studies (Computer Engineering)
Savitribai Phule Pune University

Members of Board of Studies - Computer Engineering	
Dr. Pramod Patil	Dr. Dipti Patil
Dr. Dhananjay Kshirsagar	Dr. Amol Potgantwar
Dr. Sachin Babar	Dr. Balwant Sonkamble
Dr. Suhasini Itkar	Dr. Sachin Sakhare
Dr. Dipak Patil	Dr. Vandana Dhingra
Dr. Deepali Ujalambkar	Dr. Vaishali Vikhe
Dr. Pradip Jawandhiya	Dr. Sandeep Deshmukh

Program Specific Outcomes (PSO)

- **PSO1:** Professional Skills-The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexities.
- **PSO2:** Problem-Solving Skills- The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
- **PSO3:** Successful Career and Entrepreneurship- The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

Programme Educational Objectives (PEO)

Program Educational Objectives are broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.

PEO	PEO Focus	PEO Statements
PEO1	Core competence	Attainment of key principles and practices of computation, mathematics and basic principles of engineering to ensure that graduates are able to apply their software development skills in design and implementation of practical systems consisting of software and/or hardware components.
PEO2	Problem solving skills and Ethics	Analyze real-life problems and impart science-based engineering education to develop professional skills that will prepare the students for immediate employment in the industry.
PEO3	Professionalism and Lifelong Learning	Imbibe lifelong learning, professional and ethical attitude for embracing global challenges and make positive impact on environment and society.

Knowledge and Attitude Profile (WK)

A Knowledge and Attitude Profile (KAP), often represented as WK (Knowledge and Attitude Profile) in some contexts, is a framework or assessment tool used to evaluate an individual's knowledge and attitudes related to a specific area, topic, or domain.

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

Reference: Self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) - (August 2024) Page 55.

Programme Outcomes (PO)

Program Outcomes are statements that describe what students are expected to know and be able to do upon graduating from the program. These relate to the skills, knowledge, attitude and behaviour that students acquire through the program. On successful completion of B.E. in Computer Engineering, graduating students/graduates will be able to:

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

PO10	Project Management and Finance	Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO11	Life-Long Learning	Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Reference: Self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) - (August 2024) Page 56.

General Rules and Guidelines

- **Course Outcomes (CO):** Course Outcomes are narrower statements that describe what students are expected to know, and are able to do at the end of each course. These relate to the skills, knowledge and behaviour that students acquire in their progress through the course.
- **Assessment:** Assessment is one or more processes, carried out by the institution, that identify, collect, and prepare data to evaluate the achievement of Program Educational Objectives and Program Outcomes.
- **Evaluation:** Evaluation is one or more processes, done by the Evaluation Team, for interpreting the data and evidence accumulated through assessment practices. Evaluation determines the extent to which Program Educational Objectives or Program Outcomes are being achieved, and results in decisions and actions to improve the program

Guidelines for Examination Scheme

Theory Examination: The theory examination shall be conducted in two different parts Comprehensive Continuous Evaluation (CCE) and End-Semester Examination (ESE).

Comprehensive Continuous Evaluation (CCE) :

1. CCE of 30 marks based on all the Units of course syllabus to be scheduled and conducted at institute level.
2. Case studies included under each unit are intended to support applied learning and are part of Comprehensive Continuous Evaluation
3. These case studies will be assessed through internal assessment components such as presentations, assignments, or group discussions. They shall not be included in the End-Semester Theory Examination.
4. To design a Comprehensive Continuous Evaluation scheme for a theory subject of 30 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr.	Parameters	Marks	Coverage of Units
1	Unit Test	12 Marks	Units 1 & Unit 2 (6 Marks/Unit)
2	Assignments / Case Study	12 Marks	Units 3 & Unit 4 (6 Marks/Unit)
3	Seminar Presentation / Open Book Test/ Quiz	06 Marks	Unit 5

5. CCE of 15 marks based on all the Units of course syllabus to be scheduled and conducted at institute level. To design a CCE scheme for a theory subject of 15 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr.	Parameters	Marks	Coverage of Units
1	Unit Test	10 Marks	Units 1 & Unit 2 (5 Marks/Unit)
2	Seminar Presentation / Open Book Test/ Assignments/Case Studies	05 Marks	Units 3 & Unit 4

- **Unit Test**
  - **Format :** Questions designed as per Bloom's Taxonomy guidelines to assess various cognitive levels (Remember, Understand, Apply, Analyze, Evaluate, Create).
  - **Implementation**: Schedule the test after completing Units 1 and 2. Ensure the question paper is balanced and covers key concepts and applications.
- **Sample Question Distribution**
  - Remembering (2 Marks): Define key terms related to [Topic from Units 1 and 2].
  - Understanding (2 Marks): Explain the principle of [Concept] in [Context].
  - Applying (2 Marks): Demonstrate how [Concept] can be used in [Scenario].
  - Analyzing (3 Marks): Compare & contrast [Two related concepts] from Units 1 and 2.
  - Evaluating (3 Marks): Evaluate the effectiveness of [Theory/Model] in [Situation].
- **Assignments / Case Study** : Students should submit one assignment or one Case Study Report based on Unit 3 and one assignment or one Case Study Report based on Unit 4.
  - **Format:** Problem-solving tasks, theoretical questions, practical exercises, or case studies that require in-depth analysis and application of concepts.
  - **Implementation:** Distribute the assignments or case study after covering Units 3 and 4. Provide clear guidelines and a rubric for evaluation.
- **Seminar Presentation:**
  - **Format:** Oral presentation on a topic from Unit 5, followed by a Q&A session.
  - **Deliverables:** Presentation slides, a summary report in 2 to 3 pages, and performance during the presentation.
  - **Implementation:** Schedule the seminar presentations towards the end of the course. Provide students with ample time to prepare and offer guidance on presentation skills.
- **Open Book Test:**
  - **Format:** Analytical and application-based questions to assess depth of understanding.
  - **Implementation:** Schedule the open book test towards the end of the course, ensuring it covers critical aspects of Unit 5.
- **Quiz :**
  - **Format:** Quizzes can help your students practice existing knowledge while stimulating interest in learning about new topic in that course. You can set your quizzes to be completed individually or in small groups.
  - **Implementation:** Online tools and software can be used create quiz. Each quiz is made up of a variety of question types including multiple choice, missing words, true or false etc
- **Example Timeline for conducting CCE:**
  - Weeks 1-4 : Cover Units 1 and 2
  - Week 5 : Conduct Unit Test (12 marks)
  - Weeks 6-8 : Cover Units 3 and 4

- Week 9 : Distribute and collect Assignments / Case Study (12 marks)
- Weeks 10-12 : Cover Unit 5
- Week 13 : Conduct Seminar Presentations or Open Book Test or Quiz (6 marks)

- **Evaluation and Feedback:**
  - **Unit Test:** Evaluate promptly and provide constructive feedback on strengths and areas for improvement.
  - **Assignments / Case Study:** Assess the quality of submissions based on the provided rubric. Offer feedback to help students understand their performance.
  - **Seminar Presentation:** Evaluate based on content, delivery, and engagement during the Q&A session. Provide feedback on presentation skills and comprehension of the topic.
  - **Open Book Test**: Evaluate based on the depth of analysis and application of concepts. Provide feedback on critical thinking and problem-solving skills.

End-Semester Examination (ESE)

End-Semester Examination (ESE) of 70 marks written theory examination based on all the unit of course syllabus scheduled by university. Question papers will be sent by the University through QPD (Question Paper Delivery). University will schedule and conduct ESE at the end of the semester.

- **Format and Implementation :**
  - **Question Paper Design** : Below structure is to be followed to design an End-Semester Examination (ESE) for a theory subject of 70 marks on all 5 units of the syllabus with questions set as per Bloom's Taxonomy guidelines and 14 marks allocated per unit.
  - **Balanced Coverage**: Ensure balanced coverage of all units with questions that assess different cognitive levels of Bloom's Taxonomy: Remember, Understand, Apply, Analyze, Evaluate, and Create. The questions should be structured to cover:
    * Remembering: Basic recall of facts and concepts.
    * Understanding: Explanation of ideas or concepts.
    * Applying: Use of information in new situations.
    * Analyzing: Drawing connections among ideas.
    * Evaluating: Justifying a decision or course of action.
    * Creating: Producing new or original work (if applicable).
  - **Detailed Scheme for 70 Marks :** Unit-Wise Allocation (14 Marks per Unit): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.
  - **Detailed Scheme for 35 Marks :** Unit-Wise Allocation (08 Marks for Unit 1 , 09 Marks for Unit 2, Unit 3 and Unit 4) : Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.

***

Curriculum Structure - Semester V

Third Year Engineering (2024 Pattern) – Computer Engineering

Course Code	Course Name	Course Type	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tut	Practical	CCE	ESE	Term Work	Practical	Oral	Total	Theory	Tut	Practical	Total
PCC301COM	Artificial Intelligence	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC302COM	Computer Networks	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC303COM	Theory of Computation	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC304COM	Artificial Intelligence Lab	PCC		-	2	-	-	25	25	-	50	-	-	1	1
PCC305COM	Computer Networks Lab	PCC		-	4	-	-	50	-	25	75	-	-	2	2
PEC321COM	Elective - I	PEC	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC322COM	Elective - I Lab	PEC		-	2	-	-	50	-	-	50	-	-	1	1
MDM331COM	Robotics and Automation	MDM		2	2	-	-	50		-	50	-	2	1	3
	Open Elective*	OE	2	-	-	15	35		-	-	50	2	-	-	2
ELC342COM	Technical Seminar	ELC		-	2	-	-	-	-	25	25	-	-	1	1
Total			**14**	**2**	**12**	**135**	**315**	**175**	**25**	**50**	**700**	**14**	**2**	6	**22**

***Note:** Students can opt for Open Electives offered by different discipline/faculty like Arts, Science, Commerce ,Management, Humanities or Inter-Disciplinary studies. Example – **Open Elective I** - IPR and Cyber Laws ,Sustainability Development, Digital Personal Data Protection, The Constitution of India

Programme Elective - I	
PEC321ACOM	Data Warehousing and Data Mining
PEC321BCOM	Cloud Computing
PEC321CCOM	Mobile Computing
PEC321DCOM	Embedded systems

Curriculum Structure - Semester - VI

Third Year Engineering (2024 Pattern) – Computer Engineering

Course Code	Course Name	Course Type	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tut	Practical	CCE	ESE	Term Work	Practical	Oral	Total	Theory	Tut	Practical	Total
PCC351COM	Machine Learning	PCC	4	-	-	30	70	-	-	-	100	4	-	-	4
PCC352COM	Software Engineering and Modelling	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC353COM	Machine Learning Lab	PCC	-	-	4	-	-	25	50	-	75	-	-	2	2
PCC354COM	Software Engineering and Modelling Lab	PCC	-	-	2	-	-	25		25	50	-	-	1	1
PEC361COM	Elective II	PEC	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC362COM	Elective III	PEC	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC363COM	Elective III Lab	PEC		-	4	-	-	25	50	-	75	-	-	2	2
MDM371COM	Green Computing	MDM		2	2	-	-	25	-	25	50	-	2	1	3
VSE372COM	Solar Technology and Maintenance	VSE	-	-	2	-	-	50	-	-	50	-	-	1	1
Total			**13**	**2**	**14**	**120**	**280**	**150**	**100**	**50**	**700**	**13**	**2**	**07**	**22**

Programme Elective - II	
PEC361ACOM	Quantum Computing
PEC361BCOM	Distributed System
PEC361CCOM	UX/UI Design
PEC361DCOM	System Programming

Programme Elective - III	
PEC362ACOM	Information Retrieval
PEC362BCOM	Data Visualization and Analytics
PEC362CCOM	Cryptography and Network Security
PEC362DCOM	Advanced Databases

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Computer Engineering

Semester - V



Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC301COM- Artificial Intelligence		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses : Data Structures

Companion Course: Artificial Intelligence Lab

Course Objectives:

- To introduce fundamental concepts of Artificial Intelligence, intelligent agents, and ethical considerations in AI.
- To develop the ability to model and solve problems using state-space search and heuristic algorithms.
- To enable students to design solutions for adversarial and constraint satisfaction problems using appropriate AI techniques.
- To impart knowledge representation skills using propositional and first-order logic with inference mechanisms.
- To equip students with AI planning techniques and relate them to real-world industrial applications.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain fundamental concepts of Artificial Intelligence, Agents and ethical implications of AI in real-world scenarios.
- **CO2:** To formulate real-world problems as state-space search models and implement appropriate heuristic, local, and online search algorithms
- **CO3:** To model real-world competitive and constraint-based problems and implement adversarial search algorithms and CSP techniques
- **CO4:** To represent real-world knowledge using propositional and first-order logic and apply Inference techniques and derive logical conclusions in knowledge-based AI systems.
- **CO5:** To apply AI planning techniques for solving classical planning problems and Relate them to real-world industrial AI applications.

Course Contents

Unit I - Introduction to AI and Intelligent Agents (09 hours)

Introduction to Artificial Intelligence, Foundations of Artificial Intelligence, History of Artificial Intelligence, Limits of AI, Ethics of AI, Future of AI, AI Components, AI Architectures, Intelligent Agents, Agents and Environments, Good Behavior: Concept of Rationality, Types of Agents, Nature of Environments, Structure of Agents.

Case Study: Autonomous Taxi Agent – Waymo One, AI in Healthcare – IBM Watson for Oncology

Unit–II: Problem Solving : State Space Approach and Search Strategies (09 hours)

State Space Search: Tower of Hanoi. Informed (Heuristic) Search Strategies: Introduction to Greedy BFS, A* Search, Iterative-deepening, Heuristic Functions. Local Search and Optimization Problems: Hill-climbing search, Simulated annealing, Local beam search. Online Search Agents and Unknown Environments: Online search problems.

Case Study: Warehouse robots (Amazon Kiva) and self-driving cars, Logistics and Routing: Traveling Salesman Problem, Google DeepMind – AI for Energy Efficiency in Data Centers

Unit–III: Adversarial Search and Game theory (09 hours)

Optimal Decisions in Games, Heuristic Alpha–Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs

Case Study: AlphaGo – AI in Strategic Board Games, Strategic Decision-Making in Imperfect-Information Games-- Libratus, Adversarial Search and Constraint Reasoning in Computer Chess - IBM deep blue

Unit-IV : Knowledge representation using Logical Formalisms, Propositional and First order Predicate Calculus (09 hours)

Introduction to Logical Formalisms: Role of logic in Artificial Intelligence, Knowledge-based agents Syntax and semantics of logical systems

Propositional Logic – Basics and Inference: Propositional symbols and well-formed formulas, Logical connectives, Inference rules: Modus Ponens, Modus Tollens

First Order Predicate Logic – Fundamentals, Motivation for First Order Logic, Quantifiers ($\forall$, $\exists$), Well-formed formulas, Translating natural language into FOL , Inference in First Order Predicate Logic, Unification , Resolution.

Case Study: Medical Expert System – MYCIN, Knowledge-Based Reasoning in Intelligent Search Systems:- AI-Based Rule Engine in Google Search

Unit V : Planning and Industrial Applications of AI (09 hours)

Planning: Overview, An example Domain The Blocks world, The components of planning system, Goal stack planning, Nonlinear planning using constraint posting, Hierarchical planning

Industrial Applications of AI : AI in Healthcare, AI in Finance, AI in Retail, AI in Agriculture, AI in Education, AI in Transportation, AI in Experimentation and Multi-disciplinary research

Case Study: AI-Driven Supply Chain & Production Planning (Manufacturing)

Learning Resources

Text Books:

1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN :10: 0136042597
2. Elaine Rich,Kevin Knight and Nair,"Artificial Intelligence",TMH,ISBN-978-0-07-008770-5
3. Saptarsi Goswami, Amit Kumar Das and Amlan Chakrabarti, "AI for Everyone – A Beginner's Handbook for Artificial Intelligence", Pearson, 2024
4. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN : 978-1-25-902998-1

Reference Books

1. Patrick Henry Winston, "Artificial Intelligence", Addison-Wesley Publishing Company, ISBN: 0-201-53377-4

2. Dr.Nilakshi Jain,“Artificial Intelligence,As per AICTE: Making a System Intelligent”,Wiley publication, ISBN: 9788126579945

3. Nilsson Nils J , “Artificial Intelligence: A new Synthesis”, Morgan Kaufmann Publishers Inc. San Francisco, CA, ISBN:978-1-55-860467-4

4. Dr. Lavika Goel, “Artificial Intelligence: Concepts and Applications”, Wiley publication, ISBN:97881265

e-Books:

1. http://repo.darmajaya.ac.id/5094/1/Lecture-AI.pdf

2. https://www.freebookcentre.net/ComputerScience-Books-Download/Digital-notes-on-Artificial-Intelligence

3. https://www.kdnuggets.com/10-free-artificial-intelligence-books-for-2025

MOOC :

- Artificial Intelligence: Knowledge Representation And Reasoning By Prof. Deepak Khemani, IIT Madras https://onlinecourses.nptel.ac.in/noc26_cs63/preview

- An Introduction to Artificial Intelligence By Prof. Mausam, IIT Delhi https://nptel.ac.in/courses/106102

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC302COM- Computer Networks		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses : Digital Electronics and Logic Design, Discrete Mathematics, Computer Organization & Microprocessor

Companion Course: Computer Network Lab

Course Objectives:

1. Introduce fundamental concepts of networking, hardware, software, and reference models.
2. Develop knowledge of physical, data link, network, transport, and application layers with emphasis on design issues, services, and protocols.
3. Equip students with the ability to analyze and compare routing, error control, congestion control, and quality of service mechanisms.
4. Familiarize students with widely used protocols such as TCP, UDP, IP, HTTP, DNS, SMTP, and emerging technologies in multimedia and wireless networks.
5. Strengthen problem-solving skills through case studies and exemplars that connect theoretical concepts to real-world applications.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain the fundamental concepts of computer networks, layered architecture, and physical transmission media.
- **CO2:** Apply error detection, correction, and reliable data transfer techniques at the data link layer.
- **CO3:** Analyze routing algorithms, addressing schemes, and Internet protocols at the network layer.
- **CO4**: Compare and evaluate transport layer protocols (TCP, UDP, SCTP, RTP) and mechanisms for congestion control and QoS.
- **CO5**: Demonstrate understanding of application layer protocols (HTTP, DNS, Email, FTP, TELNET, DHCP, SNMP) and emerging network applications.

Course Contents

Unit I - : Introduction to Computer Networks and Physical Layer (09 Hours)

Introduction to computer networks; uses of computer networks – business applications, home applications, mobile users ; network hardware – PAN, LAN, MAN, WAN and internetworks; network software – protocol hierarchies, design issues for layers, connection-oriented and connection-less services, service primitives, relationship between services and protocols; reference models – OSI and TCP/IP models. Physical Layer: guided transmission media; wireless transmission; telephone system; narrowband and broadband communication systems.

Case Study: Comparison of OSI vs TCP/IP in real-world networks, case study on broadband vs narrowband communication in India.

Unit–II: Data Link Layer (9 Hours)

Data Link Layer – services provided to the network layer, framing and addressing; design issues of the data link layer – error control, flow control and reliable data transfer; error detection and correction techniques – parity, checksum, CRC and basic error correction concepts; data link layer protocols – elementary protocols and Stop-and-Wait protocol; sliding window protocols – pipelining, Go-Back-N and Selective Repeat protocols; example data link layer technologies – packet over SONET

Case Study: Case study on CRC error detection in Ethernet, SONET backbone deployment in telecom networks.

Unit–III: Network Layer (9 Hours)

Network Layer – Services & Design Issues: connectionless service, connection-oriented service, QoS support, error control, flow control, store-and-forward switching, congestion control, reliability, internetworking challenges; Routing Algorithms: shortest path routing, flooding, distance vector routing, link state routing; Internet Architecture & Protocols: IP addressing (IPv4 classes, CIDR, subnetting), IPv4 vs IPv6, IP datagram, ICMP, ARP, RIP, OSPF

Case Study: Case study on IPv6 adoption in ISPs, routing comparison between RIP and OSPF in enterprise networks.

Unit-IV: Transport Layer Protocols (9 Hours)

Process to Process Delivery, Services, Socket Programming. Elements of Transport Layer Protocols: Addressing, Connection establishment, Connection release, Flow control and buffering, Multiplexing, Congestion Control. Transport Layer Protocols: TCP and UDP, SCTP, RTP, Congestion control and Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless networks.

Case Study: Case study on TCP congestion control in 4G/5G networks, RTP in video conferencing applications.

Unit V: Application Layer (9 Hours)

Introduction – principles of application layer, client-server and peer-to-peer models; Web and HTTP – request/response, persistent vs. non-persistent connections, cookies, caching, performance; DNS – hierarchy, resource records, name resolution, caching, security issues; Email – SMTP, MIME, POP3, IMAP, webmail, message format, security; FTP – basics, file transfer process, legacy relevance; TELNET – remote login, limitations, legacy use; DHCP – dynamic host configuration, IP allocation, management; SNMP – network management, monitoring, MIBs, security considerations; Emerging Topics – multimedia applications, RTP for streaming, peer-to-peer applications, cloud-based services.

Case Study: Case study on DNS security attacks (DNS spoofing), HTTP caching in CDNs, SMTP in enterprise email systems.

Learning Resources

Text Books:

1. Andrew S. Tanenbaum, David J. Wetherall – Computer Networks, 5th Edition, Pearson Education.
2. Behrouz A. Forouzan – Data Communications and Networking, 5th Edition, McGraw Hill.
3. William Stallings – Data and Computer Communications, 10th Edition, Pearson Education.

Reference Books

1. Kurose, James F., Ross, Keith W. – Computer Networking: A Top-Down Approach, 8th Edition, Pearson.
2. Peterson, Larry L., Davie, Bruce S. – Computer Networks: A Systems Approach, 5th Edition, Morgan Kaufmann.

e-Books:

1. https://people.cs.clemson.edu/~jmarty/courses/kurose/KuroseCh1-2.pdf
2. http://eti2506.elimu.net/Introduction/Books/Data Communications and Networking By Behrouz A.Forouzan.pdf
3. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf
4. https://www.tutorialspoint.com/data_communication_computer_network/data_communicati on_comp

MOOC :

1. nptel.ac.in/courses/106/105/106105183
2. nptel.ac.in/courses/106/105/106105080
3. nptel.ac.in/courses/106/105/106105081
4. nptel.ac.in/courses/106/106/106106091
5. nptel.ac.in/courses/106/105/106105031
6. https://www.mooc-list.com/tags/computer-networking
7. https://www.coursera.org/courses?query=computer%20network

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC303COM- Theory Of Computations		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses : Discrete Mathematics

Course Objectives:

1. Recall and understand the basics of mathematical concepts and machines
2. To design deterministic/nondeterministic automata and interconversion.
3. Design regular expressions and prove non-regular languages using the Pumping Lemma and Myhill–Nerode Theorem.
4. Design CFGs, PDAs, and Turing Machines for given languages.
5. Understand the basic concepts of computability theory.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply the knowledge of basics of mathematics and logic for designing Finite Automata and its variants.
- **CO2**: Construct regular expression to present regular language and understand pumping lemma and Myhill–Nerode Theorem for RE
- **CO3**: Design Context Free Grammars and learn to simplify the grammar.
- **CO4**: Construct appropriate computational models to solve given problems using Pushdown Automaton model
- **CO5**: Able to design Turing Machines for various computational problems and understand different classes of problems, classify and analyze them

Course Contents

Unit I - : Introduction to Formal Languages and Finite Automata (09 Hours)

Basic Concepts: Finite and infinite set, Symbols, Strings : Empty String, Substring of a string, Concatenation of strings, Language :Formal Language Definition, Finite representation of languages , Operations on languages: Union, Concatenation, Kleene star and Kleene plus, Concept of Basic Machine.

FA without output: Finite State Machines(FSM), Deterministic and Nondeterministic FA(DFA & NFA), epsilon NFA, Conversion of NFA with epsilon moves to NFA, Conversion of NFA to DFA, and Conversion of NFA with epsilon moves to DFA, Minimization of DFAs.

FA with output: Moore and Mealy machines - Definition, Construction, Inter- Conversion.

Case Study: FSM for vending Machine, Spell checker, Finite Automata in ATM PIN Validation System

Unit–II: Regular Expressions and Languages (09 Hours)

Introduction, Operators of RE, Precedence of operators, Algebraic laws for RE, Language to Regular Expressions, Equivalence of two REs, Kleene's theorem. Conversions: RE to NFA, DFA, DFA to RE

using Arden's theorem, Pumping Lemma for Regular languages, Closure(union, intersection, complementation, concatenation, Kleene closure) and Decision properties of Regular languages(Membership, Emptiness,Finiteness and Infiniteness) The Myhill–Nerode Theorem

Case Study: RE for variable name validation, RE to match a specific word from given string ,Regular Expressions in Email Validation System

Unit–III: Context Free Grammars (CFG) and Languages (09 Hours)

Formal Definition of Context Free Grammar, Sentential form, Derivation and Derivation Tree/ Parse Tree, Context Free Language (CFL), Ambiguous Grammar, writing grammar for language, Simplification of CFG: Eliminating $\in$-productions, unit productions, useless production, useless symbols, and Normal Forms- Chomsky normal form, Greibach normal form, Pumping Lemma for CFG, Closure properties of CFL, Chomsky Hierarchy, Applications of CFG:-Palindromes, Parenthesis Match ,Parser, Markup languages, XML and Document Type Definitions.

Case Study: Grammar Design for Arithmetic Expressions, Designing a Grammar for a Simple Programming Language Construct

Unit-IV: Push Down Automata (09 Hours)

Introduction, Formal definition of PDA, Equivalence of Acceptance by Final State & Empty stack, Non-deterministic PDA (NPDA), PDA & Context Free Language, Equivalence of PDA and CFG, PDA vs CFLs. Applications of PDA,Introduction to Post Machine,

Case Study: Applying PDA for Top-Down Parsing, Bottom-up Parsing Pushdown Automata (PDA) in Compiler Syntax Checking,XML / HTML Tag Validation

Unit V: Turing Machine and Computability Theory (09 Hours)

Turing machine (TMs): Basic model, definition, and representation, TM Instantaneous Description, Transition Function, Language accepted TM, Deterministic Turing Machines (DTM), and Construction of DTM. Universal Turing Machine (UTM), Church-Turing hypothesis, Comparison between FA, PDA and TM. Turing Machine Halting Problem.

Decidable Problems and Undecidable Problems, Church-Turing Thesis.

Reducibility: Undecidable Problems that are recursively enumerable, A Simple Undecidable.

Complexity Classes: Time and Space Measures, The Class P, Examples of problems in P, The Class NP, Examples of problems in NP, P Problem Versus NP Problem, NP-completeness and hard Problems.

Case Study: Application of Turing Machine for Language Recognition, Comparative Study of Variants of Turing Machines, Analysis of the Halting Problem

Learning Resources

Text Books:

1. Hopcroft J., Motwani R., Ullman J., "Introduction to Automata Theory, Languages andComputations", Third edition, 2008, Pearson Education Asia. ISBN: 9788131720479.

2. Michael Sipser, "Introduction to The Theory of Computation", Third edition, 2017 Thomson Course Technology, ISBN: 9781131525296.

Reference Books

1. Daniel Cohen., "Introduction to Computer Theory", Second edition, 2011, Wiley Publications (India) ISBN: 9788126513345.

2. H.R. Lewis, C. H. Papadimitriou, "Elements of the Theory of Computation", Second edition, 2006, Prentice Hall Inc. ISBN: 8131703878.

3. John C Martin. "Introduction to Language and Theory of Computation", Third edition, 2012, Tata McGraw- Hill, ISBN: 978007660489.

4. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.

5. Vivek Kulkarni, “Theory of Computation”, Oxford University Press, ISBN 0-19-808458

e-Books:

1. https://cglab.ca/~michiel/TheoryOfComputation/TheoryOfComputation.pdf 2.

2. https://www.cs.virginia.edu/~robins/Sipser_2006_Second_Edition_Problems.pdf

MOOC :

1. Theory of Computation - IIT Kanpur whttps://nptel.ac.in/courses/106104148

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC304COM- Artificial Intelligence Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks Practical : 25 Marks

Course Objectives:

1. To provide a comprehensive understanding of the fundamental concepts, principles, and methodologies of Artificial Intelligence.
2. To equip students with the ability to design and implement core AI algorithms for problem-solving, search, reasoning, and learning.
3. To develop analytical and practical skills for modeling and solving real-world problems using appropriate AI techniques.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain and differentiate fundamental concepts, components, and techniques of Artificial Intelligence, including intelligent agents and AI methodologies.
- **CO2**: Design and implement AI algorithms for state-space search, heuristic search, reasoning, and basic learning tasks using programming tools.
- **CO3**: Model real-world problems as AI problems and apply appropriate AI techniques to obtain optimal or near-optimal solutions.
- **CO4**: To apply AI planning techniques for solving classical planning problems and relate them to real-world industrial AI applications.

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

List of Assignments (Part A) - Any FOUR

1. Design and implement a simple reflex agent for the Vacuum Cleaner world and analyze its rational behavior in a given environment.
2. Implement a model-based intelligent agent capable of navigating a grid environment with obstacles using internal state representation.
3. Implement A* Search for 8 puzzle problem
4. Implement Adversarial Search with Alpha-Beta pruning
5. Implement Tower of Hanoi by State Space Search
6. Implement a Sudoku solver using backtracking and constraint propagation techniques.
7. Implement Local Beam Search for The Traveling Salesman Problem (TSP).

List of Assignments (Part B) - Any FOUR

1. Use BFS to solve a planning problem. Examples can be 8 puzzle problem, Robot path planning, Blocks world etc.
2. Use DFS for problem solving. Examples can be water jug problem , Missionaries and Cannibals ,maze solving, blocks world etc.
3. Implement Backtracking search to solve the 8-Queens constraint satisfaction problem.
4. Develop a simple chatbot using predefined rules and pattern matching.
5. Design an intelligent agent that guesses a number using feedback (higher/lower).
6. Implement Goal Stack Planning for solving the Blocks World problem and achieve a specified goal configuration.
7. Develop a game-playing agent using the Minimax algorithm for optimal decision-making.

Part C - Mini-Project (In Team of 3-4 Students) - Any ONE but not limited to

1. Medical Diagnosis System (Rule-Based Expert System) using forward and backward chaining
2. Develop a simple rule-based recommendation system (e.g., movie or book suggestion system).

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC305COM- Computer Networks Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	Term Work: 50 Marks Oral: 25 Marks

Prerequisite Courses : Digital Electronics and Logic Design, Discrete Mathematics, Computer Organization & Microprocessor, Internet of Things

Companion Course: Computer Network

Course Objectives:

1. Provide hands-on experience with computer networking concepts.
2. Familiarize students with network models, protocols, and tools.
3. Develop skills in analyzing and simulating network behavior.
4. Encourage practical implementation of routing, addressing, and error detection techniques.
5. Build competence in using tools like Packet Tracer and Wireshark for network configuration and analysis.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Compare and analyze OSI and TCP/IP reference models, topologies, transmission media, and LAN/WAN configurations using simulation tools.
- **CO2**: Apply error detection and correction techniques, subnetting, and IP addressing schemes to design efficient and reliable networks.
- **CO3**: Simulate and implement transport layer protocols (Go-Back-N, Selective Repeat, TCP/UDP sockets) and client-server applications such as file transfer and DNS lookup.
- **CO4**: Configure and evaluate routing protocols and services while capturing and analyzing traffic
- **CO5**: Work collaboratively on a mini-project to design, implement, and analyze a networking solution, demonstrating innovation, teamwork, and professional skills in laboratory practice.

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by

Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

List of Assignments (Part A) - 1, 2 mandatory and Any THREE from 3 to 7	
1	Case study on comparison of OSI and TCP/IP reference models with respect to layers, functions, protocols, and real-world implementation. (Compulsory)
2	Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool. (Compulsory)
3	Setup a WAN which contains wired as well as wireless LAN by using a packet tracer tool. Demonstrate transfer of a packet from LAN 1 (wired LAN) to LAN2 (Wireless LAN).
4	Capture and analyze Ethernet frames to study physical addressing using Wireshark Packet Analyzer Tool.
5	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.
6	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.
7	Write a program to implement link state /Distance vector routing protocol to find suitable path for transmission.

List of Assignments (Part B) - 8,9,10 mandatory and Any THREE from 11 to 15	
8	Implement Subnetting and Test IP Addressing Schemes using a packet tracer tool. (Compulsory)
9	Write a program for DNS lookup. Given an IP address as input, it should return URL and vice Versa. (Compulsory)
10	Capture and analyze HTTP, DNS, and SMTP traffic using Wireshark Packet Analyzer Tool. (Compulsory)
11	Configuration of 3 router networks using one of the following protocols RIP/OSPF/BGP Using packet Tracer tool.
12	Write a program using TCP socket for wired network for File transfer
13	Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines.
14	Configure DHCP server-client setup using a packet tracer tool.
15	Write a program to analyze following packet formats captured through Wireshark for wired Network. 1. Ethernet 2. IP 3.TCP 4. UDP

List of Assignments (Part C) - Any ONE from 16 to 19	
16	Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer Tool.
17	To study the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.).
18	Illustrate the steps for implementation of S/MIME email security through Microsoft Office Outlook.
19	To study the IPSec (ESP and AH) protocol by capturing the packets using Wireshark tool

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC321ACOM - Data Mining and Data Warehousing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Database Management Systems
Companion Course : Elective Lab I
Course Objectives: The course aims to:

1. To understand the fundamentals of Data Mining.
2. To learn the preprocessing of the data.
3. To understand the fundamentals of data warehousing and design of the data warehouse.
4. To identify the appropriate technique for mining particular type of data.
5. To understand and apply various methods, techniques and algorithms in data mining.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1**: Understand basics of data mining and apply appropriate data preprocessing techniques.
- **CO2**: Understand basics of data warehouse and to perform logical modelling of data warehouse.
- **CO3**: Understand and apply Association Rule Mining and Clustering Techniques.
- **CO4**: Understand and apply appropriate data mining algorithm to solve the problems, and explore the patterns in the data.
- **CO5**: Understand some of the advanced data mining techniques.

Course Contents

Unit I - Introduction to Data Mining and Data Preprocessing (09 Hours)

Data Mining – Motivation and Applications, Database queries Vs DM queries Data: Data, Information and Knowledge; The KDD process, Attribute Types: Nominal, Binary, Ordinal and Numeric attributes, Types of Data: Relational, Temporal, Time-series, Spatial, Text and Multimedia, Data streams, www; Introduction to Data Preprocessing, Data Cleaning: Missing values, Noisy Data; Data integration: Redundancy and Correlation Analysis only; Data reduction: Attribute Subset Selection, Sampling; Data Discretization: Binning, Histogram Analysis, Data Discretization by Data Transformation: Min-max normalization, z-score normalization and decimal scaling; Dissimilarity of Numeric Data: Minskowski Distance and Euclidean distance

Case Study: Download ZOO and Heart Disease Datasets from UCI Machine Learning Repository and study various types of attributes and format of training data.

Unit II Data Warehouse (09 Hours)

Data Warehouse: Basic Concepts, What Is a Data Warehouse? Differences between Operational Database Systems and Data Warehouses, Data Warehousing: A Multitiered Architecture, Data Warehouse Models: Enterprise Warehouse, Data Mart, and Virtual Warehouse, Extraction, Transformation, and Loading, Metadata Repository, Data Warehouse Modeling: Data Cube and OLAP , Data Cube: A Multidimensional Data Model; Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models; Dimensions: The Role of Concept Hierarchies; Measures: Their Categorization and Computation; Typical OLAP Operations.

Case Study: Design a retail sales data warehouse to analyze customer purchases, product performance, and regional sales trends. The warehouse should use a star schema with a central Sales Fact table linked to Customer, Product, Store, and Time dimensions, enabling queries on top-selling products, seasonal demand, and promotion impacts.

Unit III - Association Rules Mining and Clustering (09 Hours)

Market basket Analysis, Frequent item set, Closed item set, Association Rules, Apriori Algorithm, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, Mining Frequent Itemset without Candidate Generation: FP Growth Algorithm.

Cluster Analysis: What Is Cluster Analysis? Requirements for Cluster Analysis; Overview of Basic Clustering Methods, Partitioning Methods: K-Means: A Centroid-Based Technique k-Medoids: A Representative Object-Based Technique; Distance Measures in Algorithmic Methods, Evaluation of Clustering, Assessing Clustering Tendency, Measuring Clustering Quality.

Case Study: Apply the Apriori Algorithm to a retail sales dataset to discover frequent itemsets and generate association rules. The goal is to identify product combinations often purchased together, enabling insights for market basket analysis, cross-selling strategies, and promotional planning.

Unit IV - Classification (09 Hours)

Classification: What Is Classification? Supervised and Unsupervised Learning, General Approach to Classification, Decision Tree Induction, Decision Tree Induction; Bayes Classification Methods: Bayes' Theorem, Naive Bayesian Classification; Rule-Based Classification: Using IF-THEN Rules for Classification, Rule Extraction from a Decision Tree, Rule Induction Using a Sequential Covering Algorithm; Lazy Learners-k-Nearest- Neighbor Classifiers; Techniques to Improve Classification Accuracy: Introducing Ensemble Methods, Bagging, Boosting.

Model Evaluation and Selection: Metrics for Evaluating Classifier Performance, Holdout Method and Random Subsampling, Cross-Validation, Bootstrap.

Case Study: Download 2-3 Dataset from UCI data repository and try to run some of the classification algorithms on workbench like WEKA.

Unit V - Introduction of advanced Data Mining Techniques (09 Hours)

Introduction to Mining techniques*: Text Mining, Data stream mining, spatial and Temporal mining techniques, web mining and Recommender systems. (*All topics to be discussed at Conceptual Level).

Mining Complex Data Types: Mining Sequence Data: Time-Series, and Biological Sequences, Mining Graphs and Networks.

Case Study: Study Stock market prediction using historical price sequences to identify trends and forecast future values.

Learning Resources

Text Books:

1. Jiawei Han, Micheline Kamber, and Jian Pie, "Data Mining: Concepts and Techniques", Elsevier Publishers Third Edition, ISBN: 9780123814791.

Reference Books

1. Paulraj Ponnian, "Data Warehousing Fundamentals", John Willey.
2. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar "Introduction to Data Mining" (2nd Edition), Pearson Education.
3. Dunham, M. H. "Data mining: Introductory and advanced topics". Pearson Education.
4. Daniel T. Larose, O.P. Wali "Data Mining and Predictive Analytics (Indian Adaptation)" Wiley Publishers, ISBN: 9789354247255, 2022.

5. Parteek Bhatia, “Data Mining and Data Warehousing: Principles and Practical Techniques” Cambridge University Press.

MOOC / NPTEL/YouTube Links

1. https://onlinecourses.nptel.ac.in/noc26_cs58/preview, Data Mining, By Prof. Pabitra Mitra, IIT Kharagpur
2. https://onlinecourses.nptel.ac.in/noc26_cs14/preview , Data Mining for Decision Making, By Prof. Varun Dutt, IIT Mandi.

E-Books

1. Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, Cambridge University Press, March 2020. ISBN: 978-1108473989.(https://dataminingbook.info/book_html/)
2. Ron Zacharski, “A Programmer’s Guide to Data Mining” (http://guidetodatamining.com/)

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC321BCOM - Cloud Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Computer Networks

Companion Course: Programming Elective Course Lab

Course Objectives: The course aims to:

1. To understand the basic concepts of cloud computing and virtualization
2. To understand the implementation of virtualization in cloud computing
3. To learn the application and security on cloud
4. To study risk management in cloud computing
5. To comprehend the modern cloud environment and emerging technologies

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1**: Comprehend basic concepts of cloud computing environment
- **CO2**: Analyze Virtualization for cloud and install Virtualization software
- **CO3**: Configure, Test and Deploy applications on Cloud
- **CO4**: Understand and Apply security in cloud applications
- **CO5:** Analyze emerging technologies in modern cloud computing

Course Contents

Unit I - Introduction to Cloud Computing (09 Hours)

Cloud Fundamentals: Definition, Importance of cloud computing, Advantages and Disadvantages of Cloud Computing, Characteristics, Categories of Clouds: Private clouds, Public clouds Cloud Service Models: SaaS, PaaS, IaaS, Cloud Architecture, Cloud Storage: Distributed Data Storage, Data management, Cloud Deployment Models

Case Study: Cloud Computing Model of Amazon

Unit II Virtualization in Cloud Computing (09 Hours)

Virtualization: What's virtualization, Benefits of Virtualization, Types of Virtualization: Processor virtualization, Memory virtualization, Full virtualization, Para virtualization, and Device virtualization, Virtual Clustering, Virtualization Architecture, Containerization and orchestration, Understanding importance of Hypervisors, Virtualization Applications, Issues with Virtualization, Virtualization and Cloud Computing: Virtualizations in Cloud, Virtual Infrastructure, CPU Virtualization, Network and Storage Virtualization

Case Study: Case Study of VMware: Full virtualization, Xen: Para Virtualization, Microsoft HyperV

Unit III Cloud Platforms and Applications - (09 Hours)

Industrial Cloud Platforms: Amazon Web Services (AWS)- AWS infrastructure, Components, Amazon Simple DB, Elastic Cloud Computing (EC2), Amazon Storage System, Amazon Database Services.

Microsoft Azure: Azure core concepts, SQL Azure, and Application Services for managed runtimes. Open Source Platforms: Overview of OpenStack, CloudStack, and Eucalyptus for private cloud deployment.

Cloud Applications:

Data-Intensive & Emerging Applications: Smart Cities & IoT: Integrating sensor data from traffic, waste management, and power grids into a centralized cloud dashboard. AI/ML in the Cloud: Case study on Google Photos (image recognition) or Alexa (Natural Language Processing) using cloud-based TPU/GPU instances. Healthcare & Biology: Gene sequencing, protein folding and ECG analysis in the cloud. Geoscience: Satellite image processing and seismic data analysis using cloud clusters.

Case Study: The Google Case Study Data Processing: The evolution from MapReduce to Dremel and BigQuery. Storage Innovation: Understanding the Google File System (GFS) and BigTable as the backbone of global search.

Unit IV - Security in Cloud Computing - (09 Hours)

Risks in Cloud Computing: Risk Management, Enterprise-Wide Risk Management, Types of Risks in Cloud Computing. Data Security in Cloud: Security Issues, Challenges, advantages, Disadvantages, Cloud Digital persona and Data security, Content Level Security. Cloud Security Services: Confidentiality, Integrity and Availability, Security Authorization Challenges in the Cloud, Secure Cloud Software Requirements, Secure Cloud Software Testing

Case Study: Cloud Security Tool: Acunetix

Unit V - Modern Cloud Environment & Emerging Technologies (09 Hours)

Future Trends in cloud Computing, Mobile Cloud, Comet Cloud, Multimedia Cloud: IPTV, Energy Aware Cloud Computing, Distributed Cloud Computing Vs. Edge Computing, Containers, Dockers, Kubernets, Pod Management

Green Cloud & Sustainability: Sustainable Cloud Architecture, Energy-efficient data centre design and carbon footprint tracking.

Case Studies on DevOps: DocuSign, Forter, Gengo

Learning Resources

Text Books:

1. A. Srinivasan, J. Suresh, “Cloud Computing: A Practical Approach for Learning and Implementation”, Pearson, ISBN: 978-81-317-7651-3
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, “Mastering Cloud Computing”, McGraw Hill Education, ISBN-13:978-1-25-902995-0

Reference Books

1. James Bond ,“The Enterprise Cloud”, O’Reilly Media, Inc. ISBN: 9781491907627
2. Dr. Kris Jamsa, “Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more”, Wiley Publications, ISBN: 978-0-470-97389-9
3. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, 2010, The McGraw-Hill.
4. Gautam Shrof, “ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications”, Cambridge University Press, ISBN: 9780511778476
5. Tim Mather, Subra K, Shahid L.,”Cloud Security and Privacy”, Oreilly, ISBN-13 978-81-8404-815-5

6. Ronald L. Krutz, Russell Dean Vines, “Cloud Security: A Comprehensive Guide to Secure Cloud Computing”, Wiley- India,2010

7. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Editors: Wile, 2011

E-Books

1. http://dhoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/CLOUD%20COMPUTING%20Princip
2. https://www.lpude.in/SLMs/Master%20of%20Computer%20Applications/Sem_2/DECAP470_CLOUD_
3. https://studytm.wordpress.com/wp-content/uploads/2014/03/hand-book-of-cloud-computing.pdf
4. https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf
5. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291r2.pdf

MOOC Courses:

1. https://onlinecourses.nptel.ac.in/noc26_cs55/preview
2. http://www.ndl.gov.in/he_document/nptel/nptel/N_C_S_A_E_C_C_A_D_S_N_I_T_C_C_536752663
3. https://onlinecourses.nptel.ac.in/noc26_cs29/preview?
4. https://onlinecourses.nptel.ac.in/noc21_cs15/preview?

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC305CCOM:Mobile Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Operating Systems, Computer Networks & Security

Companion Course: Mobile Computing Lab

Course Objectives: The course aims to:

1. To understand the fundamental concepts of mobile computing and wireless transmission.
2. To study various telecommunication systems like GSM and modern cellular standards.
3. To analyse mobile network and transport layer protocols for seamless connectivity.
4. To explore mobile databases, data dissemination, and synchronisation techniques.
5. To introduce mobile application development frameworks and security challenges.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain the evolution and architecture of mobile computing and wireless transmission.
- **CO2:** Compare various telecommunication systems including GSM, GPRS, and 5G.
- **CO3:** Analyze mobile IP and transport layer variations for mobile environments.
- **CO4:** Design data management and dissemination strategies for mobile databases.
- **CO5:** Develop basic mobile applications using modern frameworks and secure them against threats.

Course Contents

Unit I -Introduction to Mobile Computing & Wireless Transmission (09 Hours)

Concepts: Evolution of Mobile Computing, History, Middleware and Gateways, Mobile Applications and Services, Constraints in Mobile Computing.

Wireless Fundamentals: Signal propagation, Hidden and Exposed terminal problems , Near–Far problem, Multiplexing (SDM, FDM, TDM, CDM), Spread Spectrum (DSSS, FHSS), MAC Protocols: SDMA, FDMA, TDMA, CDMA.

Case Study: Analysis of any real-world mobile computing application (e.g., E-commerce mobile platforms).

Unit II Telecommunication Systems (09 Hours)

GSM Architecture: Mobile Station, Base Station System, Network and Switching Subsystem, Radio Interface, Protocols, Localization and Calling.

Management: HLR/VLR Identification algorithms, Handover Management.

Evolution: 1G, 2G, 2.5G, 3G, GPRS and EDGE architecture, Introduction to 4G LTE and 5G New Radio (NR) architecture, Open RAN (O-RAN) or Network Slicing, Introduction to 6G vision, comparison of generations.

Case Study:Comparison of Handover mechanisms in 4G vs. 5G networks.

Unit III Mobile Network , Transport Layer ,& Ad-Hoc Networks (09 Hours)

Mobile IP: IP Packet Delivery, Agent Discovery, Registration, Tunneling and Encapsulation, Optimization, Reverse Tunneling, IPv6 in Mobile Networks.

Mobile TCP: Traditional TCP vs. Mobile TCP, Indirect TCP, Snooping TCP, Fast Retransmit/Fast Recovery, MPTCP (Multipath TCP).

Ad-Hoc Networks: MANET routing protocols – DSDV (Proactive), DSR and AODV (Reactive), ZRP (Hybrid), Multicast routing (ODMRP).

Vehicular Networks: VANET vs MANET, security issues in Ad-Hoc networks.

Case Study: Design a mobile IP registration flow for a device moving across different subnets.

Unit IV Mobile Application Layer Protocols & Databases (09 Hours)

Wireless Application Protocols: File system consistency, HTTP and HTML adaptations for wireless, WAP architecture (WDP, WTLS, WTP, WSP, WAE), WML, WML Script.

Mobile Databases: Data Prefetching and Staging, Caching invalidation mechanisms, Client-server computing with adaptation, Power-aware data management.

AI Integration at the Edge: Edge AI Inference for predictive prefetching, Lightweight Machine Learning models for user behaviour analysis, On-device model adaptation.

Case Study: Study of Data Synchronization in mobile healthcare apps.

Unit V - Mobile App Development & Security (09 Hours)

Frameworks: Android and iOS Architectures, Native vs. Hybrid vs. Progressive Web Apps (PWA), Introduction to Flutter/React Native.

Security: Security models in Android, Wireless network vulnerabilities, Secure data transmission, Mobile Device Management (MDM).

Emerging Integration: Cloud, Edge, IoT integration with mobile applications.

Case Study: UPI and mobile banking applications – architecture, security mechanisms, and cloud integration for real-time transactions.

Learning Resources

Text Books:

1. Mobile Communications: Design Fundamentals by William C. Y. Lee. ISBN-13: 978-0471575214, ISBN-10: 0471575216, Publisher: Wiley; 2nd edition.

2. Wireless and Mobile Network Architectures by Yi-Bing Lin, Imrich Chlamtac. ISBN-13: 978-0471394921, ISBN-10: 0471394920, Publisher: Wiley; 1st edition.

Reference Books

1. Mobile Communications by Jochen Schiller. ISBN-13: 978-8131709306, ISBN-10: 8131709309, Publisher: Pearson/PHI; 2nd edition, 2003.

2. Fundamentals of Mobile Computing by Prasant Kumar Pattnaik, Rajib Mall. ISBN-13: 978-8120344266, ISBN-10: 8120344260, Publisher: PHI Learning Pvt. Ltd., 2012.

3. Ad Hoc Mobile Wireless Networks: Protocols and Systems by C. K. Toh. ISBN-13: 978-0130078179, ISBN-10: 0130078174, Publisher: Pearson Education; 1st edition, 2002.

4. Principles of Mobile Computing by Uwe Hansmann, Lothar Merk, Martin Nicklous, Thomas Stober.ISBN-13: 978-3540002697, ISBN-10: 3540002692, Publisher: Springer; 2nd edition.

5. Mobile Computing by Tomasz Imielinski, Hank Korth. ISBN-13: 978-0792380399, ISBN-10: 0792380397, Publisher: Springer.

MOOC / NPTEL/YouTube Links

1. Mobile Computing by Dr. Selvi Rajendran, National Institute of Technical Teachers' Training and Research, Chennai, https://onlinecourses.swayam2.ac.in/ntr26_ed82/preview

2. https://www.youtube.com/playlist?list=PLmAmHQ-_5ySyvOz6_jEsevqhPno5OV41V

Savitribai Phule Pune University **Third Year - Computer Engineering (2025 Pattern)**		
PEC321DCOM- Embedded System		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Digital Electronics & Logical Design (MDM-221 COM) ,Computer Organization & Architecture (PCC-253-COM)

Companion Course: Programme Elective Course Lab 3, PEC322- Elective Lab

Course Objectives: The course aims to:

1. To Understand the fundamentals of embedded systems and their role in modern technology.
2. To understand the implementation of the various embedded components using the embedded C program.
3. To Understand the fundamentals of ARM-based systems, including architecture and its units like registers, debug interface, stack, MPU, Interrupts etc
4. To Use the various instructions to program the ARM controller.
5. To Understand the embedded system's real-time operating system and its application in IoT

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1:To Understand and Analyze various types of Embedded systems & its fundamental concepts including their architecture, components, and operational principles.
- CO2:To Understand the basic hardware components and their selection method based on the characteristics and attributes of an embedded system
- CO3: To Describe & Apply the architectural features and instructions of 32-bit microcontroller ARM Cortex M3.
- CO4: To Design & Apply the knowledge gained for Programming ARM Cortex M3 for different applications.
- CO5: To Analyze & evaluate the need of real time operating system for embedded system applications.

Course Contents

Unit I - Introduction to Embedded Systems (09 Hours)

Overview of Embedded Systems, Processor Embedded into a system, Embedded Hardware Units and Devices in system, Embedded Software, Complex System Design, Design Process in Embedded System, Formalization of System Design, Classification of Embedded Systems.

Case Study: Any one Case study based on "Healthcare", Industrial Automation" or any relevant topic which will be useful for society.

Unit II Embedded System Design Concepts (09 Hours)

Characteristics and Quality Attributes of Embedded Systems, Operational and non-operational quality attributes, Embedded Systems-Application and Domain specific, Hardware Software Co-Design

and Program Modeling (excluding UML), Embedded firmware design and development (excluding C language).

Case Study: Any one Case study based on "Smart temperature control system" or "Smart Energy Meter" or any relevant topic which will be useful for society.

Unit III ARM Architecture (09 Hours)

Introduction, RISC design philosophy, ARM design philosophy, Embedded system hardware – AMBA bus protocol, ARM bus technology, Memory, Peripherals, Embedded system software – Initialization (BOOT) code, Operating System, Applications. ARM Processor Fundamentals, ARM core dataflow model, registers, current program status register, Pipeline, Exceptions, Interrupts and Vector Table, Core extensions. Operating modes of ARM (User, FIQ, IRQ, Supervisor, Abort, Undefined and System mode)

Case Study: Any one Case study based on "Smart Home Lighting Controller" or "Automotive Engine Monitoring System" or any relevant topic which will be useful for society.

Unit IV - Introduction to ARM Instruction Set (09 Hours)

Introduction, Data processing instructions, Load – Store instruction, Software interrupt instructions, Program status register instructions, Loading constants, ARMv5E extensions, Conditional Execution

Case Study: Any one Case study based on "Embedded Operating System Service Call" or "Audio Signal Processing" or any relevant topic which will be useful for society

Unit V - RTOS and IDE For Embedded System (09 Hours)

.Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread preemption, Preemptive Task scheduling techniques, Task Communication, communication protocol ,CAN protocol for Automotive Industry, MODBUS for Industrial Automation, Ethernet for IoT application and UART, SPI, I2C for basic hardware communication.Task synchronization issues in RTOS (Racing and Deadlock) Design parameters for RTOS, Integration and testing of Embedded hardware and firmware, Embedded system Development Environment – Block diagram (excluding keil).

Case Study: Any one Case study based on "Automotive lane departure warning system" or "Robotic Arm Controller" or any relevant topic which will be useful for society.

Learning Resources

Text Books:

1. Embedded Systems - Architecture Programming and Design – Raj Kamal, 2nd ed., 2008, TMH.
2. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education
3. Andrew N Sloss, Dominic System and Chris Wright, "ARM System Developers Guide", Elsevier, Morgan Kaufman publisher, 1st Edition, 2008

Reference Books

1. Embedded Microcomputer Systems, Real Time Interfacing – Jonathan W. Valvano – Brookes / Cole, 1999,Thomas Learning

E-Books:

1. Introduction to Embedded Systems,A Cyber-Physical Systems Approach by Edward Ashford Lee, Sanjit Arunkumar Seshia

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC322COM- Elective - I Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 50 Marks

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction - Data Warehousing and Data Mining

List of Assignment - Group A (Any THREE)

1. Consider an organization of your choice. Identify a set of business processes such as sales, orders, and marketing. Design suitable data warehouse schemas including Star Schema and Snowflake Schema for analyzing these processes. Create a Fact Constellation Schema by integrating multiple fact tables. Extract data from different data sources, apply suitable transformations such as data cleaning and integration, and load the data into destination tables using an ETL tool. https://www.kaggle.com/datasets For Example: Business Origination: Sales, Order, and Marketing Process.

2. Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques (minimum 2). Visualize the clusters using suitable tool.

3. Consider a transactional dataset. Apply the Apriori algorithm to find frequently occurring itemsets. Generate strong association rules using appropriate support and confidence thresholds. Analyze the discovered rules and interpret their significance. For Example: Market Basket Analysis for identifying frequently purchased item combinations. https://www.kaggle.com/datasets

4. Consider a suitable text dataset. Perform preprocessing steps such as stop word removal, stemming, and feature selection. Represent documents using vector space models such as TF-IDF. Apply classification techniques to categorize documents and evaluate the performance using metrics such as precision and recall.For Example: Classification of news articles into categories.

5. Consider a labeled dataset belonging to an application domain such as healthcare or finance. Apply suitable data preprocessing steps such as handling missing values, normalization, and discretization. Build classification models using different techniques (minimum three), such as Logistic Regression, Decision Tree, and K-Nearest Neighbors. Evaluate and compare the models using confusion matrix and cross-validation techniques. For Example: Healthcare domain for predicting diseases.

List of Assignment - Group B - (Any TWO)

1. OLAP Operations and Data Cube Analysis Example: Sales data analysis based on time, region, and product dimensions

2. Data Preprocessing and Transformation Techniques Example: Preprocessing customer dataset for classification

3. Comparative Study of Data Mining Tools Example: Comparison of WEKA, RapidMiner, and Python libraries

List of Assignment - Group C [One Mini-Project (in group of 3 to 4 students)

1. Consider an organization belonging to an application domain such as Retail / E-commerce / Banking. Identify relevant business processes such as sales, orders, and customer transactions. Design suitable data warehouse schemas including Star Schema and Snowflake Schema for analysis of these processes. Further, integrate these schemas to form a Fact Constellation Schema. Extract data from multiple heterogeneous data sources and apply appropriate ETL (Extract, Transform, Load) operations such as data cleaning, transformation, and integration. Load the processed data into the destination warehouse tables. Generate analytical reports using suitable tools to support decision-making. For Example: Retail domain for analyzing sales trends and customer purchasing behavior.

2. Consider a suitable dataset belonging to an application domain such as Healthcare (Disease Prediction), Market Basket Analysis, or Sentiment Analysis. Perform necessary data preprocessing steps such as handling missing values, data cleaning, normalization, and transformation. Apply different data mining techniques including clustering for grouping data instances, classification for prediction of class labels, and association rule mining to discover relationships among data items. Evaluate the performance of the applied models using appropriate metrics and visualize the results using suitable tools.For Example: Healthcare domain for predicting diseases and analyzing patient data patterns.

Guidelines for Laboratory Conduction - Cloud Computing

List of Assignment - Group A (Any THREE from 1 to 5, 6th and 7th is mandatory)

1. To study fundamental concepts of cloud, understand components of Cloud Architecture and Deployment models for Cloud.
2. To study the concept of Storage as a Service (SaaS) and implement cloud storage using a cloud platform.
3. Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows 7 or 8.
4. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.
5. Implement procedure to transfer the files from one virtual machine to another virtual machine
6. Google App Engine (GAE) Setup: Install Google App Engine. Create hello world app or any other simple web applications using python/java.
7. Case Study on PaaS (Google App Engine). Use GAE launcher to launch the web applications.

List of Assignment - Group B (Any THREE from 1 to 5, 6th and 7th is mandatory)

1. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
2. Creating an application in Salesforce.com using Apex Programming Language
3. To study creating a warehouse application in Salesforce.com
4. To study installation and Configuration of Hadoop.
5. Install Hadoop single node cluster and run simple application like wordcount.
6. To study the Cloud Computing Security Issue
7. Case study on Amazon or Microsoft Azure Cloud

List of Assignment - Group **C [One Mini-Project**

1. Setup your own Cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. Write your own code for Cloud Controller using Open Source Technologies to implement with HDFS.
2. Implement the basic Operations such as divide the file in segments /blocks and download file from cloud in encrypted form. Host a portfolio or documentation site using Amazon S3 (AWS), Azure Blob Storage, or Google Cloud Storage. Integrate a Content Delivery Network (CDN) like AWS CloudFront to learn about global content distribution and HTTPS.

Guidelines for Laboratory Conduction - Mobile Computing

List of Assignment - Group A (Core Mobile Computing & Development)

1. Environment Setup: Installation of Android Studio / Flutter SDK. Create a "Hello World" app and explore the project structure, emulators, and ADB.
2. UI & Layouts: Design a multi-screen mobile application (e.g., a Profile Manager) using Constraint Layout, Intent, and Fragments.
3. Local Persistence: Create an app to perform CRUD operations on a local SQLite or Room database with data encryption.
4. Network APIs: Develop an app that fetches real-time data (e.g., Weather or News) using RESTful APIs and displays it using a Recycler View.
5. Location Services: Implement a native application that uses GPS location information to display the user's current coordinates on a Google Map.
6. Data Synchronization (SyncML concepts): Implement an Offline-First architecture where data is stored locally and synchronized with a Firebase/cloud backend once internet connectivity is restored.
7. Edge AI Integration: Integrate a pre-trained TensorFlow Lite or ML Kit model into an app for on-device image classification or text recognition (Edge AI Inference).
8. Protocol Simulation: Create a simulation to show the working of a 3G/4G/5G mobile network or an Ad-hoc network (DSDV/AODV) using NS-3 or a similar tool.

Group B: Mini Projects (Applied Learning)

1. Smart Banking App: Create an application using spinners and intents with forms for account creation, money deposit, and transaction history. Predictive
2. Prefetching App: Develop a news reader app that uses a simple on-device ML model to predict which categories the user likes and pre-fetches data accordingly (Data Prefetching and Staging).
3. Network Performance Monitor: Create a module for collecting cellular mobile network performance parameters (Signal Strength, Nearest Base Station) using the Telephony Manager API.

Guidelines for Laboratory Conduction - Embedded systems

Conduct the following experiments on an ARM CORTEX M3 evaluation board to learn Assembly Language Program and using evaluation version of Embedded 'C' & Keil uVision-4 tool/compiler.

Group A (Any FOUR Assignments)

1. Write an Assembly Language Program (ALP) to
   (a) Multiply two 16-bit numbers.
   (b) Add two 32-bit numbers.
2. Write a program to find the factorial of a number.
3. Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM.
4. Write a program to find the largest or smallest number in an array of 32 numbers.
5. Write a program for task synchronization in RTOS with the use of Mutex & Semaphore

Group B (Any FOUR Assignments)

1. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
2. Interface a DAC and generate Triangular and Square waveforms.
3. Display the Hex digits 0 to F on a 7-segment LED interface, with a suitable delay in between.
4. Interface a simple Switch and display its status through Relay, Buzzer and LED
5. Write a program for task synchronization in RTOS with the use of binary and counting semaphore, Priority Inversion.

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
MDM331COM- Robotics and Automation		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 02 Hour/Week **Practical**: 02 Hours/Week	01 01	Term Work : 50 Marks

Prerequisite Courses: Engineering Physics, Engineering Mathematics, Basics of Electrical Engineering, Basics of Electronics Engineering & Engineering Graphics.

Course Objectives: The course aims to:

1. To introduce various types of Robots and the functional elements of Robotics.
2. To impart knowledge of robot drive systems & educate on various sensors used in Robotic automation.
3. To introduce various types the end effectors.
4. To impart knowledge of basics of Robot Programming and robotic Applications

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: UNDERSTAND basic concepts of robotics.
- CO2: SELECT appropriate drive & sensors for Robotic applications.
- CO3: To COMPARE and SELECT robot and end effectors as per application.
- CO4: To know about the fundamentals of robot programming and applications.

Course Contents

Unit I - Fundamentals of Robotics (03 Hours)

Definitions of Industrial Robot, Type and Classification of Robots, Asimov‘s laws of robotics, Robot configurations, Robot Components, Robot Degrees of Freedom, Work volume and work envelope, Robot Joints and symbols, Robot Coordinates

Unit II - Sensors (03 Hours)

Transducers and sensors, Sensors in robotics, Principles and applications of the following types of sensors- Proximity Sensors, Photoelectric Sensors, Position sensors – Piezoelectric Sensor, LVDT, Resolvers, Encoders – Absolute and Incremental: - Optical, Magnetic, Capacitive, pneumatic Position Sensors, Range Sensors- Range Finders, Laser Range Meters, Touch Sensors, Force and torque sensors

Unit III - Industrial Automation and AI in Robotics (03 Hours)

Introduction to industrial automation
PLC basics and SCADA overview
IoT in automation systems
Basics of AI in robotics (computer vision, ML concepts)
Case studies: smart factories, autonomous robots

Unit IV Autonomous Navigation & Path Planning (03 Hours)

Environment Representation: Occupancy grids, topological maps, and configuration spaces.
Path Planning Algorithms: Dijkstra’s Algorithm, A* (A-star) search, and RRT (Rapidly-exploring Random Trees).
Localization: Odometry, sensor fusion (Kalman Filters), and particle filters.

Obstacle Avoidance: Potential field methods and reactive control loops.

Unit V Robotic Middleware & Industry 4.0 (03 Hours)

Robot Operating System (ROS): Architecture (Nodes, Topics, Services, Messages) and Workspace management.

Simulation Environments: Working with Gazebo and RViz for testing and validation.

Industrial Automation: Introduction to PLC (Programmable Logic Controllers) and SCADA systems.

Industry 4.0: Impact of IoT, Big Data, and AI in smart manufacturing and automation.

Learning Resources

Text Books:

1. Industrial Robotics – Mikell P. Groover, McGraw Hill
2. Robotics: Control, Sensing, Vision and Intelligence – McGraw Hill
3. Introduction to Robotics – Tata McGraw Hill
4. Robotics Engineering – Prentice Hall
5. Programmable Logic Controllers – McGraw Hill

Reference Books

Guidelines for Laboratory Conduction

- Perform all experiments using the prescribed virtual laboratory platforms before attempting hardware implementation, if available
- Carefully study the objective, theory, circuit diagram, and program before starting the experiment.
- **Ensure Arduino libraries required for simulation are available. And Internet availability.**
- Create a personal account on the required simulation platforms (Tinkercad/PLC Simulator).
- Do not modify the reference circuit initially. First execute the provided design and verify the output. And save simulation with appropriate filename on your account.
- After successful execution, modify the circuit or program to observe changes in system behavior. Compile the Arduino program and remove all errors before simulation.
- Record all observations in the laboratory journal.
- Capture screenshots of:
  - Circuit connections
  - Implemented Code (if Any)
  - Simulation observations/results
  - Serial monitor (where applicable)
  - Ladder logic program (for PLC experiment)
- Explain the working principle and demonstrate the simulation during laboratory evaluation.

List of Assignment - Group A (Any SEVEN)

1. Study of different robotic components, joints, links, and end effectors
   (a) https://www.tinkercad.com/things/c6HKUJfWCDg-keyboard-controlled-robot-arm
2. Interfacing and control of DC motor using Arduino
   (a) https://www.tinkercad.com/things/llEFwFIEoMs-arduino-dc-motor
   (b) https://www.tinkercad.com/things/4rfB2MviY4N-arduino-simple-dc-motor-control
   (c) https://www.tinkercad.com/things/lSMaDtvrRX7-arduino-dc-motor-control-learning-demo
3. Servo motor position control using PWM signals
   (a) https://www.tinkercad.com/things/77wKzsIfKp0-practical-14-to-interface-servo-motordc-motor-with-arduino-and
4. Obstacle detection robot using ultrasonic sensor
   (a) https://www.tinkercad.com/things/7SfAOdX5SrS-obstacle-avoidance-robot-using-h-bridge-and-ultrasonic-sensors
   (b) https://www.tinkercad.com/things/3gylNSXyJSu-obstacle-avoidance-robot-using-ultrasonic-sensor
5. Design and implementation of line follower robot
   (a) https://www.tinkercad.com/things/81eV0gN98mz-line-following-robot-using-ide-code
   (b) https://www.tinkercad.com/things/2c9k08J6Eod-line-follower-robot
   (c) https://www.tinkercad.com/things/hQ3SW1MiuRL-5-sensor-line-follower-robot
6. Interfacing IR sensors and proximity sensors with microcontroller
   (a) https://www.tinkercad.com/things/0n2VuaSIEZV-ir-sensor
   (b) https://www.tinkercad.com/things/7sr7qUwvkTt-proximity-sensor
   (c) https://www.tinkercad.com/things/hiofwYZLOGJ-
7. Stepper motor control for robotic arm movement
   (a) https://www.tinkercad.com/things/auKPwNpfNS3-
   (b) https://www.tinkercad.com/things/0IObHqkIYdi
8. Basic PLC programming using ladder logic for industrial automation
   (a) https://app.plcsimulator.online/
9. IoT-based automation system using sensors and cloud monitoring
   (a) https://iotdunia.com/tinkercad-arduino-simulation-tutorial/
   (b) https://pianalytix.com/weather-monitoring-system-using-tinkercad/

Savitribai Phule Pune University		
Open Electives		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	02	CCE : 15 Marks End-Semester: 35 Marks

Open Electives (OE) are multidisciplinary courses allowing students to study subjects outside their core discipline to foster holistic development and skill enhancement. Students pick subjects outside their core specialization from the following list to broaden their knowledge base.

Sr.	Open Elective Course Name	Offering Discipline
1	IPR and Cyber Laws	Law / Faculty of Humanities
2	Agri Business Management: Banking Operation and finance	Commerce & Management
3	Product Costing for Mechanical Engineering	Commerce & Management
4	Sustainability Development	Commerce & Management
5	Material and Logistics	Management
6	The Constitution of India	Law / Faculty of Humanities
7	Digital Personal Data Protection	Law / Faculty of Humanities
8	Environmental Law	Law / Faculty of Humanities
9	Construction Law and Contracts	Law / Faculty of Humanities
10	Human Resource Management	Management
11	Statistics and Computer Applications	Commerce & Management
12	Business Administration	Commerce & Management
13	Business Marketing	Commerce & Management
14	Entrepreneurship Development	Commerce & Management
15	Banking ,Finance & Insurance	Commerce & Management
16	Cost & Works Accounting	Commerce & Management

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
ELC342COM: Technical Seminar		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Oral/Presentation : 25 Marks

The "Technical Seminar" course is a critical milestone designed to bridge the gap between structured classroom learning and independent research. It prepares students for their Final Year Projects by cultivating literature retrieval, critical analysis, and technical communication skills.

Course Objectives: The course aims to:

1. To develop research orientation and technical communication skills in emerging Computer Engineering and Artificial Intelligence domains.
2. To enable students to critically review, analyze, and synthesize contemporary research papers, white papers, patents, and technical standards.
3. To promote interdisciplinary thinking aligned with NEP-2020 multidisciplinary philosophy.
4. To inculcate ethical awareness, sustainability perspective, and societal impact analysis of AI systems.
5. To prepare students for industry, higher education, entrepreneurship, and innovation ecosystems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Identify and select emerging and relevant technical topics through literature survey.
- **CO2:** Analyze and synthesize information from research papers, journals, and credible sources.
- **CO3:** Demonstrate effective technical communication skills through oral presentation.
- **CO4:** Prepare a structured technical report following academic writing standards.
- **CO5:** Use modern tools (presentation software, plagiarism checkers, referencing tools) for seminar preparation.

Guidelines for Conduct of Technical Seminar

The Technical Seminar shall be research-oriented and domain-specific, focusing strictly on recent development in Computer Engineering, Artificial Intelligence and Data Science .

Guidelines for Technical Seminar

- Topic must be from emerging Computer Engineering, Artificial Intelligence and Data Science domains (preferably last 3–5 years).
- Must involve a minimum of 5 recent research papers (IEEE, ACM, Elsevier, Springer etc).
- Student should summarizing paper – Reading abstracts and finding ideas, conclusion, advantages of their approaches and the drawbacks of the papers. They should understand the results from a research paper to related research problems. comparing the approach - Identify weaknesses and strengths in recent research articles in the subject.
- Practical sessions on how to read, analyze and summarize research papers should be arranged.
- Should not be a basic textbook topic.

- Must include: Problem statement, State-of-the-art analysis, Comparative study, Ethical & societal impact, Interdisciplinary themes aligned with NEP encouraged.
- Topic should be approval by a faculty panel/ Seminar coordinator/ Head of the department.

Seminar Process

- **Orientation, Topic Finalization and Guide allocation (Week 1–2)**
  - Conduct an orientation session explaining: Objectives of the technical seminar, Evaluation criteria and expected outcomes
  - Each student must submit: Title of the seminar, Problem statement, Relevance to current technology trends, Approval by guide is mandatory before proceeding
  - Students are assigned a faculty guide based on specialization alignment (e.g., AI/ML, Cybersecurity, Cloud Computing, IoT). Students identify 2–3 broad domains of interest.
  - The precise topic is finalized and approved by a departmental seminar committee.
- **Literature Survey & Problem Understanding (Week 2–4)**
  - Students must: Refer minimum 5–8 recent research papers (IEEE, Springer, Elsevier, ACM etc.)
  - Use scholarly databases like IEEE Xplore, Google Scholar, ScienceDirect
  - Prepare a literature survey matrix, including:
    - Author/year
    - Methodology used
    - Key findings
    - Limitations
  - Identify: Research gaps and Challenges in existing approaches
- **Synopsis Preparation & Presentation (Week 4–5)**
  - Submit a 2–3 page synopsis including: Introduction, Literature insights, Objectives, Proposed seminar scope
  - Conduct a Synopsis Presentation (5–7 minutes): Evaluate clarity of understanding, Receive feedback for improvement
  - Approval required before proceeding to full report
- **In-depth Study & Content Development (Week 5–8)**
  - Students should: Deeply analyze concepts, models, architectures, or case studies, Include diagrams, flowcharts, and comparative tables
  - Weekly review meetings with guide: Track progress, Ensure conceptual clarity,
  - Emphasis on: Critical analysis (not just description), Real-world applications
- **Draft Report Submission & Review (Week 8–10)**
  - Submit first draft of the report
  - Guide provides feedback on: Technical content quality, Structure and coherence, Referencing and plagiarism,
  - Students must revise based on suggestions by the guide

- **Pre-Seminar Presentation (Mock Evaluation) (Week 10–11)**
  - Conduct a mock presentation simulating final evaluation
  - Focus on: Presentation skills, Time management, Handling questions
  - Peer and faculty feedback should be incorporated
- **Final Report Submission (Week 11–12)**
  - Submit: Final hard copy (if required), Soft copy (PDF format)
  - Ensure:Proper formatting, Plagiarism compliance (<20%), Correct referencing using reference managers like Zotero and Mendeley Desktop
- **Final Seminar Presentation & Viva Voce (Week 12–13)**
  - Presentation duration: 10–15 minutes, Followed by Q&A session (5–10 minutes)
  - Evaluation based on: Depth of understanding, Analytical ability, Communication skills

Method of Evaluation

- During the seminar session each student is expected to prepare and present a topic on engineering/ technology, for duration of about 12to 15 minutes.
- Each student is expected to present atleast twice during the semester and the student is evaluated based on that.
- At the end of the semester, he / she can submit a report on his / her topic of seminar and marks are given based on the report.
- A Faculty guide is to be allotted and he / she will guide and monitor the progress of the student and maintain attendance.
-

Seminar Report

The final technical report should be highly structured (20 to 25 pages) , strictly technical, and clear of marketing jargon. The recommended structure includes

- **Cover Page & Certificates**: Standard institutional format.
- **Abstract:** A single, dense paragraph (200–250 words) capturing the domain background, problem identified, state-of-the-art solutions analyzed, and key takeaways.
- **Chapter 1**: Introduction: Overview of the technology, historical context, motivation, and scope.
- **Chapter 2:** Literature Survey: A comprehensive critical analysis of the referred papers. It is highly recommended to synthesize this into a structured matrix, tabular form
- **Chapter 3:** System Architecture / Technical Core: Detailed breakdown of algorithms, mathematical formulations, protocols, or structural pipelines.
- **Chapter 4:** Comparative Analysis / Applications: Deep dive into current real-world deployment challenges, performance metrics (latency, accuracy, scale), and limitations.
- **Chapter 5:** Conclusion & Future Scope: Summarizing the insights gleaned and highlighting concrete pathways for prospective final-year project extensions.

- **References:** Strictly formatted in standard IEEE

Recommended Books and References:

1. Raman, M., & Sharma, S. (2015). Technical Communication: Principles and Practice (3rd ed.). Oxford University Press.
2. Kothari, C. R. (2014). Research Methodology: Methods and Techniques (3rd ed.). New Age International Publishers.

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Computer Engineering

Semester VI



Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC351COM : Machine Learning		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 04 Hours/Week	04	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Probability and Statistics, Data Science, Python Programming
Course Objectives: The course aims to:

1. Provide the fundamental concepts of Machine Learning.
2. Develop an understanding of regression concepts, techniques, and evaluation metrics used for predictive modeling.
3. Imbibe knowledge of classification models and algorithms for solving real-world classification problems.
4. Familiarize students with clustering algorithms and ensemble learning techniques.
5. Give insight into reinforcement learning concepts and their use in sequential decision-making problems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply fundamental Machine Learning concepts in various learning paradigms and real-world engineering applications.
- **CO2:** Make use of various types of regression models for predictive modeling and data analysis.
- **CO3:** Identify different types of classification problems, including binary, multiclass, balanced, and imbalanced classification.
- **CO4:** Analyze clustering algorithms for grouping similar data points and ensemble learning techniques for improving model performance.
- **CO5:** Distinguish reinforcement learning from supervised and unsupervised learning approaches.

Course Contents

Unit I - Fundamentals of Machine Learning (12 Hours)

Introduction to machine learning, scope of machine learning, AI vs ML vs Data Science, traditional programming vs ML paradigm, and real-world engineering applications. Types of Learning: Supervised, unsupervised, semi-supervised, and reinforcement learning. Models of Machine Learning: Geometric model, probabilistic models, logical models, grouping and grading models, parametric and non-parametric models. Introduction to Feature Engineering. Feature Transformation: Dimensionality reduction techniques- Principal Component Analysis (PCA); Linear Discriminant Analysis (LDA).

Case Study: Machine Learning Based Student Performance Prediction and Feature Engineering Analysis

Unit II - Supervised Learning-Regression (12 Hours)

Introduction to regression, need of regression, regression vs correlation. Types of regression: Univariate vs Multivariate, Linear vs Nonlinear, Simple vs Multiple, Bias–Variance Tradeoff, Overfitting and Underfitting. Regression Techniques: Simple and Multiple Linear Regression; Polynomial Regression; Decision Tree Regression, Random Forest Regression, Support Vector Regression. Regularization

Techniques: Ridge Regression (L2); Lasso Regression (L1). Evaluation Metrics: Mean Squared Error (MSE); Mean Absolute Error (MAE); Root Mean Squared Error (RMSE); R-squared (R^2).

Case Study : Comparative Study of Regression Techniques for House Price Prediction

Unit III - Supervised Learning-Classification (12 Hours)

Introduction to classification, need of classification. Types of Classification: Binary and Multiclass, Binary vs. Multiclass Classification, Balanced and Imbalanced Classification Problems. Binary Classification: Linear classification model, decision boundary. Performance Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1-Score.

Multiclass Classification: One-vs-One and One-vs-All classification techniques, multiclass confusion matrix; Per-Class Precision and Per-Class Recall; Macro, Micro and Weighted Averaging Methods.

Classification Algorithms: K-Nearest Neighbors (KNN), Linear Support Vector Machine (SVM), Soft Margin SVM. Kernel Functions in SVM: Radial Basis Function (RBF/Gaussian) Kernel, Polynomial Kernel, Sigmoid Kernel.

Case Study: Comparative Study of Classification Algorithms for Email Spam Detection.

Unit IV - Unsupervised Learning and Ensemble Learning -(12 Hours)

Introduction to clustering, need for clustering, types of clustering, Hierarchical Clustering – Agglomerative and Divisive methods, Partitioning Methods: K-Means clustering algorithm, advantages and limitations, Elbow method, Silhouette method; K-Medoids, Density-Based Clustering: DBSCAN algorithm, working mechanism, advantages and limitations. Distribution-Based Clustering: Gaussian Mixture Model. Applications, introduction to Ensemble Learning, homogeneous and heterogeneous ensemble methods, advantages and limitations. Basic Ensemble Techniques: Voting (Max Voting, Averaging, Weighted Averaging). Advanced Ensemble Techniques: Bagging and Random Forest. Boosting: AdaBoost, Gradient Boosting , Stacking.

Case Study: Customer Segmentation and Sales Prediction

Unit V - Reinforcement Learning - (12 Hours)

Introduction, need of reinforcement learning, components of reinforcement learning, comparison with supervised and unsupervised learning, applications and challenges of reinforcement learning. Markov Decision Process: Markov property, elements of MDP, episodic and continuing tasks.

Reinforcement Learning Framework: Policy, state value function, action value function, Bellman equation, optimal policy.

Reinforcement Learning Algorithms: Exploration vs Exploitation, ε-greedy strategy, dynamic programming, Q-Learning algorithm and update rule, simple reinforcement learning for game playing-Tic-Tac-Toe.

Case Study: Smart Traffic Signal Control using Q-Learning.

Learning Resources

Text Books:

1. Alpaydin, Ethem, “Introduction to Machine Learning”, 2nd Edition, MIT Press, 2014, ISBN: 978-0262028189.

2. Müller, Andreas C. and Guido, Sarah, “Introduction to Machine Learning with Python: A Guide for Data Scientists”, 1st Edition, O’Reilly Media, 2016, ISBN: 978-1-449-36941-5.

3. Mitchell, Tom M., “Machine Learning”, 1st Edition, McGraw-Hill Education, 1997, ISBN: 978-0070428072.

Reference Books

1. Flach, Peter, “Machine Learning: The Art and Science of Algorithms that Make Sense of Data,” 1st Edition, Cambridge University Press, 2012, ISBN: 978-1107422223.

2. Murphy, Kevin P., “Machine Learning: A Probabilistic Perspective”, 1st Edition, MIT Press, 2012, ISBN: 978-0262018029.

3. Shalev-Shwartz, S., & Ben-David, S., “Understanding Machine Learning: From Theory to Algorithms”, 1st Edition, Cambridge University Press, 2014, ISBN: 978-1107057135.

MOOC / NPTEL/YouTube Links

1. NPTEL Course: Introduction to Machine Learning, by Prof.Balaraman Ravindran , IIT Madras https://nptel.ac.in/courses/106106139

2. NPTEL Course: Introduction to Machine Learning, by Prof.Sudeshna Sarkar.IIT kharagpur https://nptel.a

E-Books

1. Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd Edition, Springer, 2009, ISBN: 978-0387848570. https://hastie.su.domains/ElemStatLearn/

2. Sutton, Richard S., & Barto, Andrew G., “Reinforcement Learning: An Introduction”, 2nd Edition, MIT Press, 2018, ISBN: 978-0262039246. https://www.andrew.cmu.edu/course/10-703/textbook/Bart

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC352COM- Software Engineering and Modeling		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Programming Fundamentals, Data Structures, Object-Oriented Programming Concepts

Companion Description: This course provides a comprehensive understanding of software engineering principles and software modelling techniques required for developing reliable, scalable, and maintainable software systems. The course is structured into five units: two units focusing on core software engineering concepts, two units dedicated to software modelling using UML, and one unit covering latest and advanced trends in software engineering

Course Objectives: The course aims to:

1. To understand fundamental principles and challenges of software engineering.
2. To study software process models and requirement engineering techniques.
3. To apply software modelling concepts using UML diagrams.
4. To understand project management, testing concepts.
   (a) To introduce students to modern and advanced software engineering practices.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: Explain software engineering principles and software development life cycle models.
- CO2: Analyze and document software requirements using standard practices.
- CO3: Design software systems using UML-based software modelling techniques.
- CO4: Apply project management, testing, and quality assurance techniques.
- CO5: Understand and evaluate modern and advanced trends in software engineering.

Course Contents

Unit I - Introduction to Software Engineering (09 Hours)

Software Engineering: Definition, Importance, Software Crisis, Software Myths & Reality, The Software Process: Software Process Framework, SDLC, Prescriptive Process Models: Waterfall, V-Model, Incremental, Evolutionary **Process Models:** Prototyping, Spiral, Rapid Application Development (RAD), Agile Development Models: What Is Agility? Agility and the Cost of Change, Extreme Programming, Other Agile Process Models, Overview of Scrum and Kanban.

Case Study: Development of an Online Examination System to demonstrate various Software Life Cycle Models and Software Process Structures.

Unit II Requirements Engineering (09 Hours)

Requirements Engineering, Establishing the Groundwork: Identifying Stakeholders, Recognizing Multiple Viewpoints, working toward Collaboration, Asking the First Questions, Non-functional Requirements, Traceability Eliciting Requirements: Collaborative Requirements Gathering, Usage Scenarios, Elicitation Work Products, Developing Use Cases, Building the Analysis Model, Elements of the Analysis Model,

Negotiating Requirements: Requirement Validation and Management. Software Requirements Specification (SRS)

Case Study: Requirements Engineering and SRS development for a College ERP System covering stakeholder identification, requirement elicitation, use case modelling, non-functional requirements, validation and traceability.

Unit III Software Modelling (09 Hours)

Introduction to Software Architecture, Architectural design principles and patterns, Types of architectures: Layered, Client-Server, Micro services, MVC, Architectural styles and quality attributes, Architecture

documentation (4+1 view model),

Object Modelling Using UML Basic Object-Orientation Concepts, Unified Modelling Language (UML), UML Diagrams, Use Case Model, Class Diagrams, Interaction Diagrams, State Chart Diagram, Activity Diagrams, Package Diagrams, Component Diagrams, Deployment Diagram.

Case Study: Design and UML modelling of a Library Management System to demonstrate abstraction, modularity, object-oriented design principles, and complete UML modelling including structural and behavioural diagrams.

Unit IV Software Project Management and Testing (09 Hours)

Software Project Management Complexities, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO—A Heuristic Estimation Technique, Scheduling, Organisation and Team Structures, Staffing, Risk Management, Software Configuration Management.

Software Testing, A Strategic Approach to Software Testing, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System

Testing, Debugging, Defect Life Cycle, Manual and Automated testing

Case Study: a) Software Project Planning, Estimation, Scheduling, Risk and Configuration Management for an E-Governance Citizen Service Portal using COCOMO and empirical estimation techniques 2) Testing of an Online Shopping Web Application covering manual and automated testing, test planning, strategic testing approaches, object-oriented testing, Web application testing, validation, system testing and debugging techniques.

Unit V Advanced Concepts in Software Engineering (09 Hours)

Agile and DevOps practices, CI/CD pipelines, DevSecOps and Secure Software Development, Software Architecture Patterns (Event-Driven, Serverless, Reactive Systems), AI in Software Engineering (AI- assisted development, automated testing), Low-Code / No-Code Platforms, Software Sustainability and Green Computing.

Case Study: Design and implementation of a Smart City Service Portal using Agile practices, DevOps, CI/CD pipelines, Microservices architecture, Cloud-native deployment, and reusable software components.

Learning Resources

Text Books:

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

Reference Books

1. Rajib Mall, Fundamentals Of Software Engineering, Fifth Edition, ISBN-978-93-88028-02-8

2. Pankaj Jalote, An Integrated Approach to Software Engineering, Springer.
3. Booch, Rumbaugh, Jacobson, The UML User Guide, Addison-Wesley.

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC353COM- Machine Learning Lab		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	Term Work: 25 Marks Practical: 50 Marks

Prerequisite Courses: Probability and Statistics, Data Science, Python Programming

Companion Course : Machine Learning

Course Objectives:

1. Enable students to apply feature engineering techniques, implement PCA and LDA for feature reduction, and use a multiple linear regression model with suitable open-access datasets.
2. Provide knowledge of polynomial regression, regularization techniques, and logistic regression using suitable open-access datasets.
3. Introduce Support Vector Machine classifiers, K-Means clustering, and DBSCAN using suitable open-access datasets.
4. Familiarize with ensemble learning techniques and reinforcement learning using suitable datasets and simple decision-making problems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1 :** Apply feature engineering, dimensionality reduction techniques, and multiple linear regression to prepare datasets for predictive modeling.
- **CO2 :** Demonstrate the use of polynomial regression, regularization techniques, and logistic regression for regression and classification modeling using suitable datasets.
- **CO3 :** Make use of Support Vector Machine classifiers, K-Means clustering, and DBSCAN to perform classification and clustering on suitable datasets.
- **CO4 :** Evaluate predictive modeling and simple decision-making tasks using ensemble learning techniques and the Q-Learning algorithm.

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned, Instructors may also set one assignment or mini-project that is suitable to AI & DS branch beyond the scope of the syllabus.

Operating System recommended: - 64-bit Open Source Linux or its derivatives, or Windows OS.

Programming tools recommended: - Open Source Python, Programming tools like Jupyter Notebook, Pycharm, Spyder etc.

Suggested List of Laboratory Experiments/Assignments	
1	Apply basic feature engineering techniques such as handling missing values, encoding categorical variables, and scaling numerical attributes on the suitable open access dataset to prepare a clean and model ready dataset for machine learning analysis.
2	Make use of dimensionality reduction process using PCA and LDA on a suitable open-access high-dimensional dataset and observe the influence of feature reduction on model performance and computational efficiency.
3	Develop a multiple linear regression model using a suitable real-world open-access dataset and assess its predictive capability using regression evaluation metrics such as MSE, RMSE, and R^2.
4	Demonstrate the use of polynomial regression with varying polynomial degrees on a suitable open-access dataset and observe the effect of model complexity on prediction accuracy in terms of the bias–variance trade-off.
5	Classify data using regularization techniques such as Ridge and Lasso regression on a suitable multivariate open-access dataset and observe the influence on model overfitting and multicollinearity.
6	Illustrate the performance of a binary classification model using logistic regression on a suitable open-access dataset through a confusion matrix, precision, recall, and F1-score.
7	Utilize Support Vector Machine classifiers with linear and nonlinear kernels on a suitable open-access dataset and compare classification performance using appropriate evaluation metrics.
8	Apply the K-Means clustering algorithm on a suitable open-access dataset to obtain the optimal number of clusters using the Elbow or Silhouette method.
9	Employ a density-based clustering algorithm such as DBSCAN using a suitable open-access dataset to identify the clustering structure and noise points.
10	Analyze the performance of ensemble learning techniques such as Random Forest or Boosting on a suitable open-access dataset in comparison with a baseline machine learning model.
11	Evaluate the performance of reinforcement learning using the Q-Learning algorithm for a simple game or decision-making problem (e.g., Tic-Tac-Toe or Grid World), focusing on exploration–exploitation strategies and the optimal policy.

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PCC354COM- Software Engineering Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks Oral: 25 Marks

Prerequisite Courses: Object Oriented Programming, Data Structures & Database Management Systems

Companion Course : Software Engineering and Modeling (Theory)

Course Objectives:

1. To apply software engineering principles to real-world problem statements.
2. To develop Software Requirement Specification (SRS) documents.
3. To design and model software systems using UML tools.
4. To implement testing strategies and quality assurance techniques.
5. To explore Agile practices, version control, and DevOps fundamentals.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Perform feasibility analysis and project estimation for software systems.
- **CO2:** Develop SRS document using standard IEEE format.
- **CO3:** Design UML diagrams using modeling tools.
- **CO4:** Prepare and execute software test cases and quality plans.
- **CO5:** Apply Agile methodology and version control tools in software development.
- **CO6:** Develop a mini-project using complete SDLC approach.

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Term work evaluation shall be based on:

- Timely Completion – 5 Marks
- Quality of Documentation – 5 Marks
- Tool Usage & Implementation – 5 Marks
- Innovation & Problem Solving – 5 Marks
- Mini Project Performance – 5 Marks

Guidelines for Practical Examination

Problem statement jointly decided by internal and external examiners. Evaluation criteria:

- Implementation – 40%
- Understanding – 30%
- Documentation – 20%
- Viva – 10%

Guidelines for Laboratory Conduction

- Emphasis on experiential learning and skill development. Problem-based and case-based learning approach.
- Collaborative mini-project in teams of 3–4 students.
- Use of industry-relevant tools (StarUML, Visual Paradigm, GitHub). Continuous evaluation based on innovation and implementation.

List of Assignments (Part A) - Any FIVE , 6th and 7th Mandatory

1. Study and compare Waterfall Model, V-Model, Spiral Model, and Incremental Model. Select suitable life cycle model. Identify deliverables and justify model selection.
2. Identify the stakeholders involved in the selected software system, Prepare questionnaire and Develop usage scenarios.
3. Study the concepts and practices of Scrum, Kanban, and Extreme Programming used in Agile software development. Analyze how these methodologies help in managing project tasks, improving collaboration, and delivering software incrementally. Based on a selected software system, write sample user stories to capture system requirements. Further, create a sample sprint plan and track tasks using the project management tool Jira to understand Agile workflow management.
4. Prepare IEEE-format SRS document. Include Functional and Non-functional Requirements.
5. Requirements Traceability & Validation. Prepare RTM and perform requirement validation.
6. Risk identification and mitigation plan preparation.
7. Estimate using LOC/FP and apply COCOMO. Prepare Gantt chart.
8. Agile Project Planning and Tracking using Jira Study Scrum, Kanban and Extreme Programming. Prepare sample sprint in jira. To understand and implement Agile project management practices using Jira tool.

List of Assignments (Part B) Any 3 from 1 to 5, and 6,7,8 mandatory

1. Study and understand UML modeling and draw Use case diagram. Use the Unified Modeling Language (UML) to model selected system such as a Library Management System or Online Shopping System. Analyze the system requirements and represent them using Use case diagram.Also Draw Data Flow Diagram (DFD), Entity Relationship Diagram (ER Diagram).

2. UML Structural Modelling Draw Class, Package and Component Diagrams.

3. UML Behavioral Modelling Draw Sequence, Activity and State Diagrams.

4. Architectural Design Design Layered, Client-Server and Microservices architecture using 4+1 view model.

5. Software Testing – Test Case Design Write test cases to perform Black Box and White Box testing.

6. Manual and Automated Testing Perform manual testing and automated testing using tools (Selenium) Implement a simple Selenium WebDriver code in Java/Python to open Google Chrome, navigate to Google Search, and enter text in the search box.

7. Study DevOps and CI/CD Implementation Study the architecture and core concepts of DevOps and understand how CI/CD improve software delivery. Implement a simple CI/CD pipeline by creating a project repository using Git. Configure the pipeline to automatically build and test the application whenever code changes are pushed to the repository, demonstrating the basic workflow of automated software integration and deployment.

8. AI in Software Engineering / Low-Code Platform Use AI tool for code/testing OR develop app using low-code platform.

List of Assignments (Part C) - Mini-Project (In Team of 3-4 Students)

1. Problem Identification, Feasibility Study, SRS Preparation, UML Modeling, Design Implementation (optional prototype), Testing Plan, Agile Sprint Documentation, Final Report and Demonstration from any one (but not limited to)

   (a) Library Management System

   (b) Online Food Ordering System

   (c) Hospital Management System

   (d) Learning Management System

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC361ACOM- Quantum Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Engineering Mathematics, Data Structures and Algorithm, Python Programming

Course Objectives: The course aims to:

1. To understand basics of quantum computing
2. To describe qubit representation, quantum states, and measurement models and discuss mathematics required for quantum computing
3. To apply quantum gates and circuit rules to construct simple quantum circuits.
4. To analyze multi-qubit systems and basic quantum algorithms and to understand building blocks of quantum computing and design algorithms
5. To develop and test small quantum programs using quantum simulation tools.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** To define key terms such as qubit, superposition, entanglement, and quantum gate.
- **CO2:** To explain quantum state representation using Dirac notation and Bloch sphere.
- **CO3:** To construct and simulate basic quantum circuits using standard gate sets.
- **CO4:** To analyze the working of fundamental quantum algorithms and entangled systems and identify the various quantum algorithms
- **CO5:** To implement simple quantum programs using Qiskit/Cirq and interpret results and to describe usage of tools for quantum computing.

Course Contents

Unit I - Foundations of Quantum Computing (09 Hours)

Motivation for studying Quantum Computing, Origin of Quantum Computing ,Quantum Computer vs. Classical Computer, Need for quantum computing, Classical bits vs qubits, Basic quantum mechanics concepts for computing as Superposition and Entanglement, Complex numbers and vectors, Dirac notation, Bloch sphere representation.

Case Study: Quantum vs Classical Information — Modeling a Qubit System Using Bloch Sphere

Unit II - Mathematical Foundations for Quantum Computing, Qubits and Quantum Gates. (09 Hours)

Qubit states and normalization, Superposition principle, Measurement and probability, Single qubit gates: X, Y, Z, H, Phase, Rotation gates, Matrix representation of gates, Gate operations on qubits, Matrix Algebra: basis vectors and orthogonality, inner product and Hilbert spaces, matrices and tensors, unitary operators and projectors, Dirac notation, Eigen values and Eigen vectors.

Case Study: Design and Analysis of a Single-Qubit Quantum Random Number Generator (QRNG)

Unit III Multi-Qubit Systems ,Quantum Circuits and Building Blocks for Quantum Program (09 Hours)

Architecture of a Quantum Computing platform, Details of q-bit system of information representation, Multi-qubits States, Quantum superposition of qubits (valid and invalid superposition, Multi-qubit representation, Tensor product concept, Controlled gates (CNOT, CZ) ,Entanglement and Bell states ,Quantum circuit model, No-cloning theorem ,Quantum teleportation.

Case Study: Design and Analysis of a Two-Qubit Entangled System for Quantum Teleportation

Unit IV Basic Quantum Algorithms (09 Hours)

Quantum parallelism, Deutsch algorithm, Deutsch–Jozsa algorithm, Grover search algorithm, Shor algorithm, Quantum error correction using repetition codes 3 qubit codes, Shor‘s 9 qubit error correction Code.

Case Study: Speedup Through Quantum Algorithms -From Deutsch to Grover

Unit V Quantum Programming & Applications (09 Hours)

Introduction to Qiskit / Cirq ,Writing basic quantum programs ,Circuit simulation, Result visualization, Quantum cryptography basics, Quantum Machine Learning and Quantum AI, Quantum Neural Networks, Quantum Natural Language Understanding, Quantum Cryptography, Application Domains for quantum image processing overview, Current challenges and future scope, OSS Toolkits for implementing Quantum program, IBM quantum experience, Microsoft Q Rigetti PyQuil (QPU/QVM).

Case Study: Building and Deploying a Quantum Program Using Open-Source Quantum Toolkits

Learning Resources

Text Books:

1. Michael A. Nielsen, —Quantum Computation and Quantum Information||, Cambridge University Press.
2. David McMahon, —Quantum Computing Explained, Wiley, 2008.
3. Qiskit textbook https://qiskit.org/textbook-beta/
4. Vladimir Silva, Practical Quantum Computing for Developers, 2018 .
5. Wittek, “Quantum Machine Learning (What Quantum Computing Means to Data Mining)”, Peter University of Boras, Sweden - Elsevier Publications 3.
6. Andreas Winchert, “Principles of Quantum Artificial Intelligence”,Instituto Superior Técnico - Universidade de Lisboa, Portugal - World Scientific Publishing, British Library Cataloguing-in-Publication Data

Reference Books

1. Bernard Zygelman, A First Introduction to Quantum Computing and Information,2018
2. Supriyo Bandopadhyay and Marc Cahy, —Introduction to Spintronics||, CRC Press, 2008
3. The Second Quantum Revolution: From Entanglement to Quantum Computing and Other Super-Technologies, Lars Jaeger
4. La Guardia, Giuliano Gladioli —Quantum Error correction codes Springer,2021
5. Press Stephen Kan, “MetricsandModelsinSoftwareQualityEngineering||,Pearson,ISBN-10:0133988082; ISBN-13:978-0133988086 2.
6. Michael A. Nielsen, “Quantum Computation and Quantum Information”, Cambridge University PressStephen Kan, —Metrics and Models in Software Quality Engineering, Pearson, ISBN-10: 0133988082; ISBN-13: 978-0133988086 3.

7. David McMahon, “Quantum Computing Explained”, Wiley .

8. Microsoft Quantum Development Kithttps://www.microsoft.com/enus/quantum/development-kit Forest SDK PyQuil: https://pyquil.readthedocs.io/en/stable/

9. Amazon Bracket Documentation on AWS:https://aws.amazon.com/braket/ 7 D-Wave Systems Documentation: https://docs.dwavesys.com/docs/latest/index.html

E-Books

1. https://quantum.cloud.ibm.com/learning/en

2. https://quantum.cloud.ibm.com/docs/en/guides/quick-start

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC361BCOM -Distributed Systems		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Good understanding of Programming and Problem Solving concepts
Course Objectives: The course aims to:

1. To learn the fundamentals of Distributed Systems
2. To learn types of communication and synchronization in Distributed Systems
3. To acquaint with the Distributed File Systems
4. To understand consistency and replication in Distributed Systems
5. To understand the fault tolerance based Distributed Systems
6.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Analyze Distributed Systems types and architectural styles
- **CO2:** Implement communication and Synchronization algorithm mechanism in Distributed Systems
- **CO3:** Apply replication techniques and consistency model in Distributed Systems
- **CO4:** Develop the components of Distributed File System
- **CO5:** Implement different distributed algorithms over current distributed platforms.

Course Contents

Unit I -Introduction to Distributed Systems, Distributed Architecture (09 Hours)

Definition and Characteristics, Properties of Distributed Systems relation to Computer System, Design goals: Supporting resource sharing, Making distribution transparent, Open, Scalable, Pitfalls. Distributed Computing vs other computing models, Types of Distributed Systems, Architectural styles: Layered architectures, Object-based architectures, Publish-Subscribe architectures, Middleware organization: Wrappers, Interceptors, Modifiable middleware. System architecture: Centralized, Decentralized, Hybrid, Example architectures – Network File System, Web, Architectures Design Issues and Challenges- Systems perspective, Algorithm perspective, Driven by new applications

Case Study: Case Study of Middleware System that includes Design, Architecture and Application.

Unit II Communicationand Synchronization in Distributed Systems(09 Hours)

Introduction, Layered Protocols, Types of Communication, Remote Procedure Calls, Message Oriented Communication, Stream Oriented Communication, Multicast Communication,

Clock Synchronization: Physical Clocks, Global Positioning System,Clock Synchronization Algorithms, Logical Clocks: Lamport's Logical Clocks, Vector Clocks

Mutual Exclusion: A Centralized Algorithm, A Decentralized Algorithm, A Distributed Algorithm, A Token Ring Algorithm, A Comparison of the Four Algorithms

Election Algorithm: Traditional Election Algorithms, Elections in Wireless Environments,Elections in Large-Scale Systems

bMessage Queuing Systems

Unit III Coordination, Consistency and Replication in Distributed Systems (09 Hours)

IntroductionData Centric Consistency Models: Continuous Consistency, Consistent Ordering of OperationsClient Centric Consistency Models: Eventual Consistency, Monotonic Reads, Monotonic Writes, Read Your Writes, Writes Follow ReadsConsistency Protocols: Continuous Consistency, Primary-Based Protocols, Replicated-Write Protocols,A Cache-Coherence Protocols, Implementing Client-Centric Consistency

Replication: Reasons for Replication, Replication as Scaling Technique, Replica-Server Placement, Content Replication and Placement, Content Distribution

Case Study: Services Bayou and Coda

Unit IV Distributed File Systems (09 Hours)

Introduction, Features of Distributed filesystem, File Models, File Accessing Methods, File Sharing Semantics, File Caching Schemes, File Replication, Fault Tolerance, Atomic Transaction, Design Principles, Security.

Naming: Names, Identifiers, and Addresses, Types: Flat Naming, Structured Naming, Attribute-Based Naming

Case Study: DCE Distributed File System, SunNetwork, Andrew File System

Unit V (Advancements in Distributed Systems 09 Hours)

Epidemic and gossip-based algorithms, Napster and Gnutella networks, DHTs: Chord, Pastry and BitTorrent, Logical clocks, Mutual Exclusion Algorithms, Distributed Leader Election, Distributed minimum spanning tree, The FLP result,Consistency models and the CAP theorem, Paxos and Raft, Byzantine General's Problem, Virtual synchrony, Bitcoin and Blockchains, Amazon Dynamo,

Case Study: Facebook Cassandra, Google Percolator, Voldemort (LinkedIn), Condor, and Microsoft DryadLINQ (Any one)

Learning Resources

Text Books:

1. Maarten van Steen, Andrew S. Tanenbaum, "Distributed System Principles and Paradigms", Third edition, version 3
2. George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems Concepts and Design", Fifth edition

Reference Books

1. Kshemkalyani Ajay D, Mukesh Singhal, "Distributed Computing: Principles, Algorithms andSystems", Cambridge Press, 2011.
2. Singhal and Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill
3. Vijay K.Garg, "Elements of Distributed Computing", Wiley

MOOC / NPTEL/YouTube Links

1. Prof. Rajiv Misra, Distributed System, https://nptel.ac.in/courses/106/106/106106168/#
2. Prof. Rajiv Misra, Distributed System, https://nptel.ac.in/courses/106/104/106104182/
3. Prof. Smruti R. Sarangi, Advance Distributed System, https://nptel.ac.in/courses/106102237

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC361CCOM- User Interface and User Experience Design		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to

1. To understand principles of user-centered interface design.
2. To apply visual and interaction design techniques for digital interfaces.
3. To analyze user requirements using UX research methods.
4. To develop wireframes and prototypes for applications.
5. To explore modern UI/UX tools and industry practices.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain fundamental concepts of UI/UX and user-centered design principles.
- **CO2:** Apply human-computer interaction and visual design principles in interface design.
- **CO3:** Analyze user needs using UX research techniques and information architecture.
- **CO4:** Develop wireframes and interactive prototypes using modern design tools.
- **CO5:** Evaluate modern UI/UX trends, tools, and industry practices for designing effective digital products.

Course Contents

Unit I -Introduction to UI/UX Design - (09 Hours)

What is UI/UX Design: Importance of User-Centric Design, Goals of User Interface Design , Design Thinking Process, Core Principles, Role of UX in Product Development Lifecycle, mental and conceptual model.

Usability Design Principles: 4 Design Principles, Schneiderman's Golden Rules, Gestalt Principles of Design, Visual Design Principles, Form versus Function, Metaphors, Idioms and Affordances in UI design, User Interface Elements: Input Controls, Navigation Components, Information Components, Containers User Research: Qualitative and Quantitative User Research, Behavioral and Attitudinal User Research, Use of Personas, User Stories and Scenarios, Affinity Mapping.

Case Study: A good and a bad User Interface Design

Unit II - Usability Engineering, Evaluation and Testing (09 Hours)

Usability Engineering: Concept of usability, usability principles, benefits of usability for users and organizations, internationalization and localization, human errors and their impact on usability. Usability Evaluation: Human information processing and memory, Fitts's Law and Hick's Law, usability inspection methods such as Heuristic Evaluation and Cognitive Walkthrough, user studies and field studies. Usability Testing: Planning and conducting usability testing, Think-Aloud testing, A/B testing, use of heatmaps, and basic UX metrics for evaluating user experience.

UX Design Foundations: Ideation and research in UX design, content and interaction mapping, paper prototyping, introduction to wireframes and interface layout, applying Nielsen's usability heuristics, refining UI based on user feedback.

Unit III - WEB DESIGN: STRATEGIES AND INFORMATION ARCHITECTURE (09 Hours)

User- Centric Design - The UX Phases - Waterfall vs. Agile - Web vs. App. Determining Strategy: User Research - Inspiration - Analytics - User Needs and Client Needs - Target Audience - What is in and What is Out: Outlining Scope - Content and Functionality. The Sitemap: Introduction to Sitemaps - Information Architecture - Sitemap Concerns - annotated process - Elements - Treejack Introduction - Treejack Analysis.

Unit IV - Wireframing Fundamentals - (09 Hours)

Introduction to wireframes, purpose and importance in UX design, low-fidelity vs high-fidelity wireframes, sketching interfaces, layout grids, design hierarchy, and responsive layouts. Prototyping Concepts: Introduction to prototypes, types of prototypes (paper prototypes, digital prototypes, interactive prototypes), fidelity levels, advantages of prototyping in UX design, iterative design process. Interaction Design: Principles of interaction design, micro-interactions, feedback mechanisms, navigation flows, usability considerations in interaction design, designing intuitive interfaces. UI Design Tools: Introduction to modern UI/UX design tools such as Figma, Adobe XD, Sketch, and InVision. Creating interactive prototypes, collaborative design workflows, design handoff to developers. Design Documentation: User flow diagrams, storyboards, design specifications, and design system basics.

Unit V - Designing UX for Tomorrow - (09 Hours)

Emerging technologies in UX Design: Voice UI, Touchless gesture control, Intelligent UX, Conversational UX, Immersive Media and Fluid UX Designing for Web and Mobile Interfaces, IoT applications, Industry Specific UX Design: FinTech, Education, Health Care, E-commerce and Industrial Websites, designing for Wearable Devices, designing for Augmented Reality, Virtual Reality and Mixed Reality, Tomorrow's challenges in UX design.

Learning Resources

Text Books:

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos – Designing the User Interface: Strategies for Effective Human-Computer Interaction, Pearson.
2. Don Norman – The Design of Everyday Things, Basic Books.
3. Russ Unger and Carolyn Chandler – A Project Guide to UX Design, New Riders.

Reference Books

1. Steve Krug – Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders.
2. Alan Cooper, Robert Reimann, David Cronin – About Face: The Essentials of Interaction Design, Wiley.
3. Jesse James Garrett – The Elements of User Experience, New Riders.
4. Jeff Gothelf & Josh Seiden – Lean UX, O'Reilly.

MOOC / NPTEL/YouTube Links

1. Google UX Design Professional Certificate – Coursera
2. Interaction Design Specialization – Coursera (UC San Diego)
3. User Experience Design Fundamentals – Udemy
4. Introduction to User Experience Design – Georgia Tech (Coursera)
5. Figma UI/UX Design Essentials – Udemy

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC361DCOM - System Programming		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Fundamentals of Programming Language and Data Structures
Companion Course : System Programming Lab
Course Objectives: The course aims to:

1. To understand and explain the fundamental concepts of system software, machine architecture, language processors and assemblers.
2. To introduce the concepts and design of macroprocessors, loaders, and linkers used in system software.
3. To understand lexical and syntax analysis, and learn the concepts of token recognition, parsing, syntax-directed translation, and error handling.
4. To understand the phases of a compiler, interpreter and debugger
5. To introduce key system software components and their real-world applications

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain and demonstrate the working principles of system software components, particularly assemblers.
- **CO2:** Analyze and describe the design and functioning of macroprocessors, loaders, and linkers in program execution.
- **CO3:** Design and implement basic lexical analyzers and parsers to identify tokens, recognize grammar structures, and handle simple program errors.
- **CO4:** Analyze and differentiate the various phases of a compiler, interpreter, and debugger.
- **CO5:** Explain and evaluate the roles of device drivers, system utilities, runtime systems, and virtualization tools in modern computing environments.

Course Contents

Unit I -Overview of System Software and Language Processors (09 Hours)

Fundamentals of system software and its components, difference between system and application software, basic machine architecture, instruction cycle, addressing modes, and machine/assembly language concepts. Overview of language processors including assembler, compiler and interpreter with phases of compilation and Bootstrapping, Assembler design covering symbol, literal and pool tables, directives, forward references, macro processor, and one-pass and two-pass techniques for Assembler and Macro Processor. Basics of linker, loader, system utilities, Types of errors, error detection and diagnostics.

Case Study: Development of System Programming Tools at Microsoft

Unit II Macro Processors, Linkers, Loaders (09 Hours)

Macroprocessors: Introduction, macro definition and macro call, macro expansion, nested macro calls, advanced macro facilities, functions and basic tasks of a macro processor, features and design issues of macro processors, macro processor design options, design of macro preprocessor and macro

assembler, one-pass and two-pass macro processors. **Linkers & Loaders:** Introduction to linkers and loaders, relocation and linking concepts, design of a linker, self-relocating programs, linking in MS-DOS, linking of overlay structured programs, static and dynamic linking. Loaders and loading process, different loading schemes such as compile-and-go loaders, general loader schemes, direct loaders, absolute loaders. Comparison of linkers and loaders.

Case Study: macro expansion and program relocation using a two-pass macro processor and relocating loader for executing an assembly language program.

Unit III Scanning, Parsing and Language Translator (09 Hours)

Programming Language Grammars, Classification of Grammar,Ambiguity in Grammatical Specification, Context Free Grammar. Lexical analysis: role of lexical analyzer, tokens, patterns and lexemes, lexical errors, and regular definitions for language constructs such as strings, sequences, and comments. Transition diagrams and finite automata for token recognition, reserved words and identifiers with suitable examples. Scanning, Parsing, Parsing fundamentals, syntax analysis, Top Down Parsing-LL, Bottom up Parsing-LR, syntax-directed translation concepts, and basic error handling and recovery mechanisms.

Case Study: LEX/FLEX and YACC/BISON—introduction, specification and features, working flow, and applications in compiler construction, language processing, and syntax analysis.

Unit IV Compilers, Interpreters and Debugger (09 Hours)

Compiler : Introduction, Definition, and Functions of a Compiler; Structure of a Compiler; Data Structure used in Compiling, Scope Rules, Memory Allocation,Compilation of Regular Expression, Compilation of Control Structure. Phases of a Compiler; Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code Generation, Code Optimization, and Target Code Generation.

Interpreter: Definition and working of interpreter, interpreter vs. compiler, use cases of interpreters, performance considerations, Examples: JavaScript interpreters

Debugger: Need for debugging tools, types of program errors, Debugging techniques, Overview of popular debuggers (GDB, IDE debuggers)

Case Study: GCC (GNU Compiler Collection)

Unit V - Device Drivers and Modern Extension (09 Hours)

Device Driver: Introduction, Definition ,Types :Character device drivers, Block device drivers, Network device drivers, Printer and display driver.

System Utilities : File management utilities, Disk management tools , Backup and recovery utilities, Compression utilities.

Runtime Support Systems : Runtime libraries, Garbage collectors, Just-In-Time (JIT) compilers , Virtual machines (JVM, .NET CLR)

Virtualization and Container Software : Hypervisors (VMware, VirtualBox, KVM), Container engines (Docker)

Case Study : Virtualization Using Type-2 Hypervisor: A VirtualBox

Learning Resources

Text Books:

1. D. M. Dhamdhere, System Programming and Operating Systems, McGraw Hill Education.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi & Jeffrey D. Ullman, “Compilers: Principles, Techniques, and Tools (2nd Edition)”, Pearson Education, ISBN-13: 978-0321486813.
3. System Software by Santanu Chattopadhyay, Prentice-Hall India,2007

Reference Books

1. Leland L. Beck, System Software: An Introduction to Systems Programming, 3rd Edition, Pearson Education.

2. Andrew W. Appel, “Modern Compiler Implementation in C”, Cambridge University Press, ISBN-13: 978-0521607650.

3. John J. Donovan, System Programming, Tata McGraw-Hill, ISBN-13: 978-0-07-460482-3.

MOOC / NPTEL/YouTube Links

1. NPTEL – Compiler Design https://nptel.ac.in/courses/106/105/106105190

2. Coursera – Compilers (Stanford University) https://www.coursera.org/learn/compilers

3. Udemy – Compiler Design (Beginner to Advanced) https://www.udemy.com/course/compiler-design/

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC362ACOM - Information Retrieval		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to:

1. To understand the concepts of Information Retrieval (IR) and clustering in IR.
2. To study indexing structures for Information Retrieval.
3. To evaluate the performance of IR systems and understand user interfaces for searching.
4. To understand information sharing on the web and applications of IR with emphasis on multimedia, distributed IR and web search.To explore web search, multimedia retrieval, and recent trends in Information R

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** To Understand IR concepts and apply clustering techniques in IR
- **CO2:** To Study indexing structures for Information Retrieval.
- **CO3:** To evaluate the performance of IR systems and understand user interfaces for searching
- **CO4:** Map IR concepts to recent developments in the field of Information Retrieval.
- **CO5:** Interpret recent trends and applications in Information Retrieval

Course Contents

Unit I - Introduction to IR & Text Processing (09 Hours)

Introduction to Information Retrieval – Definition, Need, IR vs Database Systems, Applications of IR, Components of an IR System, Architecture of a Search Engine, Document Representation, Types of Documents (Structured, Semi-structured, Unstructured), Text Preprocessing: Tokenization, Stop Word Removal, Stemming (Porter Algorithm), Lemmatization, Normalization Techniques, Term Weighting: Bag of Words, Term Frequency (TF), Inverse Document Frequency (IDF), TF-IDF, Vector Space Model (Basics), Cosine Similarity, Introduction to Inverted Index (Structure & Need)

Case Study: Spam Email Filtering using Text Preprocessing

Unit II - Information Retrieval Models (09 Hours)

Boolean Retrieval Model – Boolean Queries (AND, OR, NOT), Query Processing, Limitations of Boolean Model, Introduction to Ranked Retrieval, Vector Space Model (Detailed) – Document Vectors, Query Vectors, Similarity Measures, Document Ranking using Cosine Similarity, Probabilistic Retrieval Model – Basic Probability, Binary Independence Model, Relevance Feedback, BM25 (Basic Concept), Language Models for IR (Query Likelihood, Smoothing Basics) + Evaluation Metrics (Precision, Recall, F1, MAP Overview)

Case Study: Vector Space Model in Academic Digital Libraries

Unit III - Index Compression and Dynamic Inverted Indices (09 Hours)

Performance evaluation: Precision and recall, alternative measures

Ontology: Ontology based information sharing, Ontology languages for semantic web, Ontology creation.

General-Purpose Data Compression, Modeling and Coding, Huffman Coding, Arithmetic Coding, Symbol wise ,Text Compression

Compressing Postings Lists: Nonparametric Gap Compression, Parametric Gap Compression, Context-Aware Compression Methods, Index Compression for High Query Performance, Compression Effectiveness, Decoding Performance, Document Reordering.

Dynamic Inverted Indices: Incremental Index Updates, Contiguous Inverted Lists, Noncontiguous Inverted, Document Deletions: Invalidation List, Garbage Collection, Document Modifications,

Case Study: Ontology-Based Medical Information Retrieval (e.g., PubMed + MeSH Ontology)

Unit IV - Web Searching (09 Hours)

Introduction, Challenges, Web Characteristics, Search Engines: Centralized Architecture, Distributed Architecture, User Interfaces, Ranking, Crawling the web, Indices, Browsing, Meta-searchers, Searching using Hyperlinks, Trends and Research Issues, Introduction to Web Scraping: Python for web scraping, Request, HTML parsing, Beautiful Soup.

Case Study: Web Crawling Architecture (Googlebot)

Unit V - Advanced Information Retrieval (09 Hours)

XML Retrieval: Basic XML concepts, Challenges in XML retrieval, Vector space model for XML retrieval, Evaluation of XML retrieval, Text-Centric vs. Data-Centric XML retrieval. Recommendation System: Collaborative Filtering and Content Based Recommendation of Documents and Products. Introduction to Semantic Web.

Case Study: Personalized Search using User Profiling Focus Areas

Learning Resources

Text Books:

1. Yates & Neto, "Modern Information Retrieval", Pearson Education, ISBN 81-297-0274-6.
2. C.J. Rijsbergen, "Information Retrieval", (www.dcs.gla.ac.uk).
3. Heiner Stuckenschmidt, Frank van Harmelen, “Information Sharing on th Semantic Web”, Springer International Edition, ISBN 3-540-20594-2.

Reference Books

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, UniversitiesPress
2. Data Warehousing Fundament’s, Pualraj Ponnaiah, Wiley Student Edition.
3. The Data Warehouse Life Cycle Toolkit — Ralph Kimball, Wiley StudentEdition.
4. Data Mining, Vikaram Pudi, P Radha Krishna, Oxford University Press

MOOC / NPTEL/YouTube Links

1. https://onlinecourses.nptel.ac.in/noc25_cs19/preview
2. https://workshop.sps.nyu.edu/~sultans/dwdm/

E-Books

1. https://freecomputerbooks.com/Information-Retrieval-A-Survey.html?utm_source
2. https://freecomputerbooks.com/Information-Retrieval-A-Survey.html?utm_source

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC362BCOM - Data Visualization and Analytics		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to:

1. To understand the concept of Data Analytics and its Lifecycle
2. To understand computational statistics in Data Science
3. To understand and apply data modeling strategies for predictive and analytical tasks
4. To understand text analytics techniques and their applications
5. To become conversant with recent advances in analytics

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Understand fundamental concepts of Data Science, Big Data, and the Data Analytics Life cycle
- **CO2:** Apply inferential statistical techniques to analyze datasets
- **CO3:** Implement predictive analytics models using Python programming
- **CO4:**Analyze text data using text mining techniques and evaluate machine learning models using standard metrics
- **CO5:** Implement data visualization using visualization tools in Python programming

Course Contents

Unit I - Introduction to Data Analytics and Life Cycle (09 Hours)

Introduction to Data Science and Big Data, Applications of Data Science, 5 V's of Big Data, Relationship between Data Science and Information Science, Business intelligence versus Data Science, Data: Data Types, Data Collection. Need of Data wrangling, Methods: Data Cleaning, Data Integration, Data Reduction, Data Transformation, Data Discretization.

Data Analytic Life cycle: Introduction, Phase 1: Discovery, Phase 2: Data Preparation, Phase 3: Model Planning, Phase 4: Model Building, Phase 5: Communication results, Phase 6: Operationalize.

Case study: Global Innovation Social Network and Analysis (GINA).

Unit II - Inferential Statistics 09 Hours

Statistical concepts, Measures of Central Tendency: Mean, Median, Mode, Mid-range. Measures of Dispersion: Range, Variance, Mean Deviation, Standard Deviation. Bayes theorem, Basics and need of hypothesis and hypothesis testing, Pearson Correlation, Sample Hypothesis testing, Chi-Square Tests, t-test.

Case Study: For an employee dataset, create measure of central tendency and its measure of dispersion for statistical analysis of given data.

Unit III Predictive Analytics (09 Hours)

Essential Python Libraries, Basic examples. Data Preprocessing: Removing Duplicates, Transformation of Data using function or mapping, replacing values, Handling Missing Data.

Data Analytics Types: Predictive, Descriptive and Prescriptive. Association Rules: Apriori Algorithm Regression: Linear Regression, Logistic Regression. Classification: Decision Trees. Introduction to Scikit-learn: Installations, Dataset, Regression and Classification using Scikit-learn. Clustering Algorithms: K-Means.

Case Study: Use Spam mail dataset and apply data preprocessing methods.

Unit IV - Text Analytics and Model Evaluation (09 Hours)

History of text mining, Roots of text mining overview of seven practices of text analytic, Application. Introduction to Text Analysis: Text-preprocessing, Bag of words, TF-IDF and topics. Need and Introduction to social network analysis. Naïve Bayes classifier for text classification, Model Evaluation and Selection: Metrics for Evaluating Classifier Performance, Holdout Method and Random Sub sampling, Parameter Tuning and Optimization, Confusion matrix, AUC-ROC Curves.

Case Study: Use Spam mail dataset and apply Naïve Bayes classifier and evaluate model.

Unit V - Data Visualization and Big Data Processing (09 Hours)

Introduction to Data visualization, Challenges to Big data visualization, Data Visualization using Python: Line plot, Scatter plot, Histogram, Density plot, Box-plot, Pie Chart, Area Plot, Heatmap, Funnel Chart. Open source data visualization tools, Data Visualization using Tableau.

Hadoop Ecosystem and Technologies, Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands, Anatomy of File Write and Read, NameNode, Secondary NameNode, and DataNode, Hadoop MapReduce paradigm, Map Reduce tasks, Job, Task trackers, Apache SPARK Framework, Difference between Hadoop and Apache Spark.

Case Study: Sales Data Visualization using Tableau.

Learning Resources

Text Books:

1. David Dietrich, Barry Hiller, "Data Science and Big Data Analytics", EMC education services, Wiley publication, 2012, ISBN0-07-120413-X
2. Jiawei Han, Micheline Kamber, and Jian Pie, "Data Mining: Concepts and Techniques Elsevier Publishers Third Edition, ISBN: 9780123814791, 9780123814807

Reference Books

1. EMC Education Services, "Data Science and Big Data Analytics- Discovering, analyzing Visualizing and Presenting Data"
2. 2. DT Editorial Services, "Big Data, Black Book", DT Editorial Services, ISBN: 9789351197577, 2016 Edition
3. Chirag Shah, "A Hands-On Introduction To Data Science", Cambridge University Press, (2020), ISBN : ISBN 978-1-108-47244-9
4. Wes McKinney, "Python for Data Analysis ", O' Reilly media, ISBN: 978-1-449-31979-3
5. Trent Hauk, "Scikit-learn Cookbook", Packt Publishing, ISBN: 9781787286382
6. Jenny Kim, Benjamin Bengfort, "Data Analytics with Hadoop", OReilly Media, Inc., ISBN:978149191370
7. Venkat Ankam,"Big Data Analytics", Packt Publishing, ISBN: 9781785884696
8. Seema Acharya, Subhashini Chellappan, "Big Data And Analytics", Wiley publication, ISBN: 9788126579518

MOOC / NPTEL/YouTube Links

1. Computer Science and Engineering - NOC:Data Science for Engineers
2. Computer Science and Engineering - NOC:Python for Data Science
3. Computer Science and Engineering - NOC:Data Mining
4. Computer Science and Engineering - NOC:Big Data Computing
5. Big Data Computing – Course

E-Books

1. An Introduction to Statistical Learning by Gareth James https://www.ime.unicamp.br/∼dias/Intoductio
2. Python Data Science Handbook by Jake VanderPlas https://tanthiamhuat.files.wordpress.com/2018/04/
3. Introducing Data Science by Davy Ciele, Manning Publications
4. Introducing Data Science [PDF]
5. Handbook for visualizing : a handbook for data driven design by Andy krik
6. A Handbook for Data Driven Design
7. An introduction to data Science : https://docs.google.com/file/d/0B6iefdnF22XQeVZDSkxjZ0Z5VUE/ed
8. Hadoop Tutorial :
   (a) https://www.tutorialspoint.com/hadoop/hadoop_tutorial.pdf?utm_source=7_&utm_medium=aff
   (b) Learning with Python; How to think like a computer scientist:http://openbookproject.net/thinkcs/p
9. Python for everybody: http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
10. Scikit Learn Tutorial : https://scikit-learn.org/stable/

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC362CCOM - Cryptography and Network Security		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Computer Networks, Cloud Computing
Companion Course : Programme Elective Course Lab 3
Course Objectives: The course aims to:

1. To understand fundamental concepts of cryptography and network security.
2. To study classical and modern encryption techniques.
3. To learn authentication, digital signatures, and key management.
4. To acquire the knowledge of security protocol
5. To understand network security and security mechanisms in real-world systems and applications.
6.

Course Outcomes: Upon successful completion of this course, students will be able to:

- CO1: To understand the fundamental principles of cryptography and network security.
- CO2: To analyze symmetric and asymmetric encryption algorithms.
- CO3: To implement authentication and key management mechanisms.
- CO4: Identify and Evaluate information security threats and vulnerabilities in Information systems
- CO5: To study network security and evaluate security issues in modern computing environments.

Course Contents

Unit I - Introduction to Security Basics (09 Hours)

Introduction, Security concepts and need of security, Types of security attacks, security approaches, Security services, Security mechanisms, Model for network security, Basic Terminologies in Network Security, Cryptography concepts and techniques, Steganography.

Case Study: 1. Data Hiding in Military Communication Using Cryptography & Steganography, 2. Data Breach in an Online Banking System

Unit II Data Encryption Techniques (09 Hours)

Symmetric key Ciphers: Stream Ciphers, Substitution Techniques- Caeser Cipher, Monoalphabetic Ciphers, Playfair Cipher, Hill Cipher, Polyalphabetic Ciphers, Transposition Techniques, Block Ciphers and Data Encryption standards, 3DES, Advanced Encryption standard (AES)

Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange, Knapsack Algorithm, Elliptic Curve Cryptography (ECC)

Case Study: 1. Secure Online Payment System Using Symmetric & Asymmetric Encryption 2. Secure Military Communication Using Classical & Modern Ciphers

Unit III Data Integrity Algorithms and Web Security (09 Hours)

Cryptographic Hash Functions- Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA), SHA-3, MD4, MD5, Message Authentication Codes - Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MACs, Digital Signatures : Digital Signatures, Schemes, Digital Signature standard, X.509 Certificate

Web Security issues, HTTPS, SSH, Email security : PGP, S/MIME

Case Study: 1. Securing Online University Examination System 2. Secure Corporate Communication Network Using IPSec, SSH & Email Security

Unit IV Security Requirements (09 Hours)

IP Security: Introduction, Architecture, IPV6, IPv4, IPSec protocols, and Operations, AH Protocol, ESP Protocol, ISAKMP Protocol, VPN. WEB Security: Introduction, Secure Socket Layer (SSL), SSL Session and Connection, SSL Record Protocol, Change Cipher Spec Protocol, Alert Protocol, Handshake Protocol. Electronic Mail Security: Introduction, Pretty Good Privacy, MIME, S/MIME, Comparison. Secure Electronic Transaction (SET).

Case Study: 1. Secure Corporate VPN Network Using IPSec & SSL 2. Secure Online Shopping System Using Email Security & SET

Unit V Network, Wireless and Emerging Security (09 Hours)

The OSI Security architecture, Access Control, Flooding attacks, Intrusion detection, Host based and network based Honeypot, Firewall and Intrusion prevention system: Need of firewall, Firewall characteristics, Types and access policy, Intrusion prevention system, Host based, Network based, Hybrid, Application Security, Security maintenance, Multilevel Security, Concepts of trusted system.

Wireless Network Security: WEP, WPA, WPA2, WPA3, Web Attacks: XSS, SQL Injection, CSRF (Basics), Blockchain Fundamentals and Cryptographic Concepts, Cloud Security Basics, IoT Security Challenges

Case Study: 1. SQL Injection Attack on E-Commerce Website 2. Secure Banking Infrastructure with Multilevel Security & Trusted Computing

Learning Resources

Text Books:

1. William Stallings, Lawrie Brown, “Computer Security Principles and Practice”, 3rd_Edition, Pearson
2. William Stallings, “Cryptography and Network Security Principals and Practice”, Fifth edition, Pearson
3. Nina Godbole, Sunit Belapure, “Cyber Security”, Wiley, ISBN: 978-81-265-2179-1

Reference Books

1. Atul Kahate,” Cryptography and Network Security”, 3e, McGraw Hill Education
2. Prakash C. Gupta, “Cryptography and Network Security”, PHI
3. V.K. Pachghare, “Cryptography and Information Security”, PHI Learning
4. Bernard Menezes, “Network Security and Cryptography”, Cengage Learning India, 2014, ISBN No.: 8131513491

MOOC / NPTEL/YouTube Links

1. Cryptography I – Offered by Stanford University (Coursera)

2. Cybersecurity Fundamentals – Offered by University of Maryland (Coursera)
3. Network Security – Offered by Georgia Tech (Udacity / Online Platforms)
4. Recommended course of NPTEL: 1. Cryptography and Network Security – Prof. Debdeep Mukhopadhyay, IIT Kharagpur 2. Network Security – Prof. Dheeraj Sanghi, IIT Kanpur

E-Books

1. William Stallings – Cryptography and Network Security (Pearson eBook Edition)
2. Jonathan Katz & Yehuda Lindell – Introduction to Modern Cryptography (CRC Press eBook)
3. NIST Special Publications on Cryptographic Standards (Available online at NIST website)
4. RFC documents on TLS, IPSec, SSH (IETF Digital Library)

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC362DCOM - Advanced Databases		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to:

1. To understand advanced DBMS concepts, system architectures (centralized, distributed, parallel and cloud).
2. To develop knowledge of query processing techniques, cost estimation methods, and query optimization strategies in centralized, parallel, and distributed environments.
3. To acquire understanding of transaction management, concurrency control protocols, and recovery mechanisms to maintain database consistency and reliability.
4. To gain knowledge of advanced database models including parallel databases, distributed databases, and XML-based data management systems.
5. To understand database security principles, protection mechanisms, privacy techniques, and recent trends in modern database technologies.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Analyze and compare various advanced DBMS architectures.
- **CO2:** Evaluate query execution plans, estimate query costs, and design optimized query processing strategies for large-scale database applications.
- **CO3:** Analyze transaction schedules, implement appropriate concurrency control techniques, and design recovery mechanisms for real-world database systems.
- **CO4:** Design and evaluate parallel and distributed database systems and implement XML storage, validation, and querying mechanisms.
- **CO5:** Implement database security mechanisms, analyze security threats, and evaluate modern database technologies including cloud and NoSQL systems.

Course Contents

Unit I - Introduction, Basic Concepts and Advanced DBMS Architecture (09 Hours)

Introduction to Advanced DBMS : Review of basic Database Management System (DBMS) concepts, Need for Advanced DBMS, Limitations of Traditional DBMS, Applications of Advanced Database Systems

Advanced DBMS System Architecture: Centralized Architecture, Single-Site Database System, Client–Server Architecture (Two-tier Architecture & Three-tier Architecture), Role of Application Server, Advantages and Limitations of Client–Server Systems, Client–Server vs Peer-to-Peer Architecture

Data Distribution Techniques: Data Fragmentation, Data Replication

Parallel Database Architecture: Overview of Shared Memory Architecture, Shared Disk Architecture, Shared Nothing Architecture, Performance Improvement using Parallelism

Cloud Database Architecture: Introduction to Cloud Databases, Database as a Service (DBaaS), Multi-tenant Architecture, Scalability and Elasticity.

Case Study: Design and implement a Cloud-Based Hospital Management Database System. Create ER/EER diagrams and implement the schema in a DBMS. The system should support multi-branch

data management, patient records, appointments, billing, and data replication. Justify the selected database architecture (centralized/distributed/cloud/NoSQL).

Unit II Unit ll Query Processing and Optimization (09 Hours)

Query Processing : Overview , Measures of Query Cost , Selection Operation , Sorting ,Join Operation , Evaluation of Expressions , Query Processing in Memory

Query Optimization : Overview , Transformation of Relational :Expressions , Estimating Statistics of Expression : Results , Choice of Evaluation Plans , Materialized Views , Advanced Topics in Query : Optimization

Parallel and Distributed Query Processing: Parallel Evaluation of Query Plans , Distributed Query Processing

Case Study: Online Retail Management System that maintains large volumes of data related to customers, orders, products, and payments. The database contains relations such as Customer(CustID, Name, City), Orders(OrderID, CustID, OrderDate), OrderDetails(OrderID, ProdID, Quantity), and Product(ProdID, Category, Price). Management frequently executes analytical queries to generate sales reports, customer behavior insights, and inventory statistics.

Unit III Transaction Management, Concurrency and Recovery (09 Hours)

Introduction to Database Transaction, Serial and Non-Serial Schedules, Serializability – Conflict and View, Cascading Rollback, Recoverable and Non-Recoverable Schedules, Strict and Rigorous Schedules. Concurrency Control: Lock-Based Protocols, Timestamp-Based Protocols, Validation-Based Deadlock Handling. Recovery System: Failure Classification, Log-Based Recovery, Write-Ahead Logging (WAL), Checkpoints, Recovery with Concurrent Transactions, ARIES (Algorithm for Recovery and Isolation Exploiting Semantics.

Case Study: E-Commerce Order Processing System:Study of simultaneous order placement, inventory update, and payment processing transactions to illustrate concurrency control, isolation levels, and rollback mechanisms.

Unit IV Advanced Database Models (09 Hours)

Introduction to Parallel Databases - Parallel database architecture, speedup, scale-up I/O parallelism, Inter-query and Intra-query parallelism, Inter-operational and Intra-operational parallelism, parallel query evaluation, Design of parallel systems, Implementation issues of Parallel query evaluation,

Introduction to Distributed Databases- Study of DDBMS architectures, Comparison of Homogeneous and Heterogeneous Databases, Analysis of Concurrency control in distributed databases, Implementation of Distributed query processing, distributed data storage.

XML Technologies- Introduction to XML, Structure of XML Data, XML Schema and DTD, XML Validation, XPath, XQuery and FLWOR Expression, XML Storage and APIs, XML & DTD Implementation Querying and Transformation: XQuery, FLOWR, XPath, XML validation, Web server, API to XML, Storage of XML Data, XML Applications: web services, Web based system, Implementation of XML validations, Use of web servers.

Case Study: Design a Distributed Banking Database System that simulates multiple branch databases. Implement distributed queries and demonstrate parallel query execution concepts. Additionally, create and validate XML documents for customer transaction data and perform XML querying using XPath/XQuery.

Unit V Database Security and Recent Trends (09 Hours)

Fundamentals of Database Security: Importance of database security, Security challenges Security goals: Confidentiality, Integrity, Availability (CIA), Basic authentication (passwords, roles), Authorization models (privileges, role-based access control),Protection Mechanisms.

Access control techniques: Database encryption, and Column-level and full-database encryption Monitoring and Threats: Database logging and audit trails, Database Activity Monitoring (Overview), SQL Injection, Privilege escalation and insider threats.

Privacy and Compliance: Data sanitization, Data masking and anonymization,

Recent Trends and Emerging Technologies: Overview of NewSQL, Trends in modern database architectures, Cloud database security, NoSQL security concerns, AI-enhanced security mechanisms

Case Study: Develop a Secure University Database System with role-based access control (Admin, Faculty, and Student). Implement authentication, authorization, encryption of sensitive data, audit logging, and demonstrate protection against SQL injection. Include a brief study of cloud/NoSQL security features.

Learning Resources

Text Books:

1. Silberschatz, Korth, Sudarshan, Database System Concepts, 7th Edition, McGraw-Hill.

Reference Books

1. Ramakrishnan&Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill.
2. Elmasri, Navathe, Fundamentals of Database Systems, 7th Edition, Pearson.
3. Wiese, Lena, Advanced Data Management: For SQL, NoSQL, Cloud and Distributed Databases, 1st Edition, Springer.

MOOC / NPTEL/YouTube Links

1. https://nptel.ac.in/courses/106106133Database Management System ,IIT Madras / IIT Bombay-https://nptel.ac.in/courses/106105175 nptel.ac.in/courses/106106220
2. https://onlinecourses.nptel.ac.in/noc26_cs72/preview
3. https://onlinecourses.nptel.ac.in/noc26_cs09/preview

E-Books

1. File Download Database Management Systems PDF by Raghu Ramakrishnan, Johannes Gehrke
2. NoSQL Database, by Christof Strauch: FREE Book Download

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
PEC363COM - Elective III Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	**Term Work**: 25 Marks **Practical** : 50 Marks

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction - Information Retrieval

List of Assignment - Group A (Any THREE)

1. Implement a Conflation-Based Text Document Representation System.
2. Implement a Single-Pass Clustering System for grouping 4–5 text documents.
3. Implement a Document Retrieval System Using Inverted File Indexing.
4. Implement an Information Retrieval Evaluation System to calculate Precision, Recall, F-Measure, and E-Measure.
5. Implement a Case Study on an Information Retrieval Based Recommendation System (Product / Doctor / Music / Price Recommendation System

6. Implement basic PageRank algorithm for ranking web pages.

List of Assignment - Group B - Mini-Project (In Team of 3-4 Students)

1. Mini Project: Boolean Retrieval Model Implementation
2. Mini Project: Web Page Ranking using PageRank Algorithm
3. Mini Project: Clustering Documents using K-Means
4. Mini Project: Stop-word Removal and Stemming Analyzer

Guidelines for Laboratory Conduction - Data Visualization and Analytics

- Set of suggested assignment list is provided in groups - A and B.
- Each student must perform 10 assignments (8 from group A, 2 from group B), 1 mini project from Group C
- Operating System recommended :- 64-bit Open source Linux or its derivative
- Programming tools recommended: - JAVA/Python/R/Scala/Tableau

List of Assignment - Group A

1. Data Wrangling I: Perform the following operations using Python on any open source dataset (e.g., data.csv)
   (a) Import all the required Python Libraries.
   (b) Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).
   (c) Load the Dataset into pandas dataframe. create own data frame.
   (d) Find shape, head(),tail(), iloc() ,Describe(),Info() on dataframe.
   (e) Handling Missing Data (Detect missing values/non-missing Values,)
   (f) Fill missing values(with a specific value,Forwardfill method)
   (g) Find mean mode,median of specific column and used to fill missing value
   (h) Use interpolation methods to fill gaps(mean,mode,median,Standard deviation)
   (i) Turn categorical variables into quantitative variables in Python.
2. Data Wrangling II : Create an "Academic performance" dataset of students and perform the following Data Transformation operations using Python.
   (a) Replacing values
   (b) Scaling/normalizing data (Use z-score,Min-Max normalization)
   (c) Pivot tables
   (d) Melting
   (e) Stacking/Unstacking
3. Data Analytics I: Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features.
4. Data Analytics II:
   (a) Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
   (b) Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.
5. Data Analytics III

(a) Implement Simple Naïve Bayes classification algorithm using Python/R on spam mail dataset.

(b) Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

6. Text Analytics

   (a) Extract Sample document and apply following document preprocessing methods:Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.

   (b) Crate representation of document by calculating Term Frequency and Inverse Document Frequency.

7. Data Visualization I :Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.

   (a) Draw Normal Curve, line plot, scatter plot, Density and contour plots, Correlation and scatter plots.

   (b) Create a histogram for each feature in the dataset to illustrate the feature distributions.

   (c) Create a boxplot for each feature in the dataset.

   (d) Compare distributions and identify outliers.

   (e) Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

8. Data Visualization II : Visualizing geographic data using Tableau.

List of Assignment - Group B (Any TWO)

1. Implementation of Hadoop environment including installation, mode configuration, service management using startup scripts, and editing of Hadoop configuration files.

2. Implement of Word count with Hadoop Map Reduce.

3. Write a simple program in SCALA using Apache Spark framework

List of Assignment - Group C [One Mini-Project (In Team of 3-4 Students and all statements should cover)]

1. Use a customer reviews dataset and classify reviews into positive, negative, and neutral sentiments using Python and scikit-learn.

2. Use the following COVID-19 vaccination data dataset and perform following analytics on the given dataset

   - Describe the dataset
   - Number of persons state wise vaccinated for first dose in India
   - Number of persons state wise vaccinated for second dose in India
   - Number of Males vaccinated
   - Number of females vaccinated

3. Create an interactive dashboard to visualize state-wise COVID-19 vaccination progress in India. **Visualizations:**

- State-wise first dose vs second dose
- Male vs Female vaccination
- Top 10 states by vaccination
- Trend over time (line chart)
- India map (choropleth)

4. Use a credit card transaction dataset to detect fraudulent transactions.
    - Describe the dataset
    - Perform class imbalance handling
    - Train classification model
    - Evaluate model using ROC-AUC
    - Predict fraud for new transactions

Guidelines for Laboratory Conduction - Cryptography and Network Security

List of Assignment

1. Implementation of Classical Ciphers: Write a program to implement classical encryption techniques such as Caesar Cipher, Playfair Cipher, and Vigenère Cipher. Demonstrate both encryption and decryption processes and analyze their security limitations.
2. Implementation of AES / DES Algorithm: Develop a program to implement symmetric key encryption using AES and/or DES algorithms. Perform encryption and decryption of a given plaintext and compare the performance and security features of both algorithms.
3. Implementation of RSA Algorithm: Write a program to implement the RSA public key cryptographic algorithm. Generate public and private keys, encrypt a message using the public key, and decrypt it using the private key.
4. Diffie–Hellman Key Exchange: Implement the Diffie–Hellman Key Exchange algorithm to securely generate a shared secret key between two communicating parties over an insecure channel.
5. Digital Signature Generation and Verification: Develop a program to generate and verify digital signatures using appropriate cryptographic algorithms. Demonstrate message authentication and integrity verification.
6. Packet Analysis using Wireshark: Perform network packet analysis using Wireshark. Capture live network traffic, identify different protocol layers, and analyze packet details such as source/destination IP addresses and port numbers.
7. Basic Firewall Configuration: Configure a basic firewall using system tools (e.g., Windows Firewall or iptables in Linux). Create rules to allow and block specific traffic and test the configuration.
8. SSL/TLS Certificate Generation: Generate a self-signed SSL/TLS certificate using OpenSSL. Configure a local server to enable secure communication and verify the certificate functionality.
9. Hashing and HMAC Implementation: Write a program to implement cryptographic hash functions (e.g., SHA-256) and HMAC. Demonstrate message integrity verification and compare hashing with HMAC for authentication purposes.
10. Mini Project
    (a) In many fields such as military communication and confidential corporate communication, protecting sensitive information during transmission is critical. Traditional encryption secures the content but does not hide the existence of the message. Therefore, a system is required that both encrypts the message and hides it within another medium. Develop a secure communication system that encrypts a secret message using a cryptographic algorithm (such as AES or RSA) and then hides the encrypted message inside an image using steganography techniques. The receiver should be able to extract the hidden data and decrypt it to obtain the original message.
    (b) Web applications are frequently targeted by attackers through vulnerabilities such as SQL Injection, which allows unauthorized access to databases and sensitive information. Developers must understand how these attacks occur and how to prevent them. Develop a web-based application that demonstrates a SQL Injection attack scenario and then implement appropriate security measures to prevent such attacks. The project should show both the vulnerable version and the secured version using techniques such as input validation and prepared statements.

Guidelines for Laboratory Conduction - Advance Databases

List of Assignment (Any SIX)

1. To study and implement 2-tier client–server architecture and analyze their advantages and limitations.
   - Task : 2-tier architecture
     - Create a database named collegedb.
     - Create tables:
       a. Student(studentid, name, course, year)
       b. Course(courseid, coursename, credits)
     - Insert at least 5 records.
     - Develop a simple front-end program to: insert data, update data, delete data, display records
2. To explore advanced database architectures and perform basic implementation using sql
   - Task: Distributed database
     (a) Create two separate databases (site1db and site2db).
     (b) Demonstrate horizontal or vertical fragmentation.
3. Employee management database : employee(empid, name, deptid, salary) ,department(deptid, deptname, location)
   (a) Transform sql queries into optimized relational algebra expressions.
   (b) Apply selection and projection push-down rules.
   (c) Demonstrate query rewriting using a materialized view
4. Distributed sales database : sales_region1(saleid, product, amount) ,sales_region2(saleid, product, amount) ,sales_region3(saleid, product, amount)
   (a) Design a distributed query execution plan for global aggregation queries.
   (b) Explain parallel evaluation of joins and aggregations.
5. To study and implement transaction control and isolation levels in mysql and observe their effect on concurrent transactions.
6. To study and simulate log-based recovery techniques used in database systems to recover data after system failures.
7. Write a program to check if a given schedule is serial, serializable, conflict serializable.
8. Design and simulate a recovery algorithm implementing undo, redo, and undo–redo
9. Develop a secure database application to prevent sql injection attacks
10. Write a program for data encryption and monitoring in database systems

List of Assignment - Group B - Mini-Project (In Team of 3-4 Students) Any one

1. Build a small library management system that demonstrates:
   (a) Basic distributed concept

(b) Query optimization
(c) Simple concurrency
(d) Simple security mechanism

2. Develop a small retail database demonstrating:

(a) Query optimization
(b) Transaction management
(c) Basic recovery
(d) SQL injection prevention

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
MDM371COM - Green Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 02 Hours/Week **Practical**: 02 Hours/Week	02 01	Term Work : 25 Marks Oral : 25 Marks

Course Objectives: The course aims to:

1. Understand environmental impacts of computing technologies.
2. Explore sustainable hardware, software, and networking solutions.
3. Analyze energy efficiency in modern IT infrastructure.
4. Apply Green IT policies and intelligent techniques to build sustainable computing systems.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Explain environmental impacts of computing and the need for green IT solutions.
- **CO2:** Analyze green business processes, enterprise architecture, and sustainable information systems.
- **CO3:** Evaluate energy-efficient techniques used in data centers and communication networks.
- **CO4:** Apply sustainability metrics, regulatory standards, and AI-based approaches to optimize energy consumption in computing systems.

Course Contents

Unit I - Fundamentals of Green Computing (03 Hours)

Introduction to Green Computing, Definition of Green Computing, Need for Green Computing, Objectives of Green Computing, Relationship between Business, IT and Environment, Environmental Impact of Information Technology, Energy Consumption in IT Infrastructure, Electronic Waste (E-Waste), Environmental Effects of E-Waste, Holistic Approach to Green IT, Concept of Greening IT Infrastructure, Green IT Strategies for Organizations, Drivers of Green IT Adoption, Energy Efficient Computing Practices, Sustainable IT Management, Role of IT in Environmental Sustainability, Green Data Usage, Digitalization to Reduce Paper Consumption, Eco-friendly Computing Technologies, Green Procurement in IT, Green Supply Chain in IT, Benefits of Green Computing, Challenges in Implementing Green Computing.

Case Study: A university campus replaces hundreds of old computers every year, resulting in large amounts of electronic waste. Improper disposal may cause environmental pollution due to toxic materials. Propose a sustainable e-waste management system for the campus

Unit II - Green Assets and Modelling (03 Hours)

Green Assets: Buildings, Data Centres, Networks, and Devices – Green Business Process Management: Modeling, Optimization, and Collaboration – Green Enterprise Architecture – Environmental Intelligence – Green Supply Chains – Green Information Systems: Design and Development Models.

Case Study: Green Information Systems: Design and Development Model

Unit III Data Centres and Networking (03 Hours)

Data centre sustainability: Power Usage Effectiveness (PUE) metrics, Virtualization: Consolidating servers and storage, Energy-efficient networking and communication technologies, A detailed syllabus

on Green Computing focusing on Data Centres and Networking, Energy-efficient hardware, virtualization, cooling techniques, and sustainable networking protocols., Green Facility Design: Site selection, modular data centre design, building orientation, and energy-efficient building materials. Power Efficiency: Efficient power distribution (AC vs. DC), Uninterruptible Power Supplies (UPS) in high-efficiency modes, and smart PDUs., Cooling Systems Optimization: Hot/cold aisle containment, blanking panels, liquid cooling (direct-to-chip/immersion), and free cooling (outside air economization), Renewable Energy Integration: On-site solar/wind generation, Power Purchase Agreements (PPAs), and Battery Energy Storage Systems (BESS)

Case Study: Study on : AI Infrastructure (Cisco): specialized AI-ready data center to manage intensive workloads, using Nexus Dashboard for centralized control.

Unit IV Compliance, Policies, Case Studies (03 Hours)

Regulations: WEEE Directive, RoHS, and environmental laws

Metrics: Measuring, monitoring, and reporting energy consumption

AI for Sustainable Computing : AI-based energy consumption monitoring , Machine learning models for predicting power usage in data centers , AI-driven cooling systems for green data centers

AI-based Resource Optimization :Intelligent workload scheduling , Smart power management in cloud computing , AI-based demand prediction to reduce idle computing resources

Case Study: Green IT in telecom, banking, manufacturing AI-based green data centers, Green computing in homes and offices

Unit IV - AI in Green IT (03 Hours)

AI in Environmental Monitoring :AI models to measure carbon footprint , Smart IoT sensors with AI for energy efficiency , AI-enabled smart buildings and smart campuses

Green AI : Energy-efficient machine learning models, Reducing computational cost of deep learning , Model compression and efficient AI algorithms

Learning Resources

Text Books:

1. Hwaiyu Geng, Data Center Handbook, 2nd Edition, John Wiley & Sons, 2014. ISBN: 978-1118436639

2. San Murugesan, G. R. Gangadharan (Editors), Harnessing Green IT: Principles and Practices, 1st Edition, John Wiley & Sons / IEEE Press, 2012. ISBN: 978-1119970057

3. Bhuvan Unhelkar, Green IT Strategies and Applications: Using Environmental Intelligence, 1st Edition, CRC Press / Taylor & Francis, 2011. ISBN: 978-1439815977

Reference Books

1. Carl Spathis, Kevin Curran, Green Computing: Tools and Techniques for Saving Energy, Money and Resources, 1st Edition, CRC Press, 2012. ISBN: 978-1439879283

2. Jason Harris, Green Computing and Green IT: Best Practices on Regulations and Industry, 1st Edition, CreateSpace Independent Publishing, 2008. ISBN: 978-1440439612

3. Bud E. Smith, Green Data Centers: Steps for the Journey, 1st Edition, Prentice Hall / IBM Press, 2009. ISBN: 978-0137043750

4. Jorge Marx Gómez, Bernhard Niemann, Richard Thomas, Green IT: Technologies and Applications, 1st Edition, Springer, 2012. ISBN: 978-3642221958

MOOC / NPTEL/YouTube Links

1. Green Computing – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc20_cs79/preview
2. Cloud Computing – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs18/preview
3. Internet of Things – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs06/preview
4. Sustainable Engineering – IIT Madras https://onlinecourses.nptel.ac.in/noc22_ce95/preview

E-Books

1. IEEE Xplore Digital Library – Green Computing Research Papers
2. Association for Computing Machinery Digital Library – Sustainable Computing Research
3. Sustainable Computing
4. Sustainability, Green IT and Education Strategies in the Twenty-first Century

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

List of Assignment (Part A any 5, Part B any 2 and 1 Mini Project)

Part A:

1. Study of Energy Consumption of Computing Devices: Different computing devices consume different amounts of electrical energy depending on their hardware configuration and workload. Analyze the energy consumption of devices such as desktop computers and laptops under different conditions including idle state, document editing, internet browsing, and high CPU utilization. Record and compare the power usage and estimate the annual electricity consumption. Based on the analysis, recommend suitable green computing practices for reducing energy consumption in educational institutions or offices.

2. Survey and Classification of Electronic Waste (E-Waste): Rapid technological advancements lead to frequent replacement of electronic devices resulting in significant electronic waste. Conduct a survey of electronic devices available in a computer laboratory or department. Identify obsolete or unused equipment and classify them into reusable, recyclable, and hazardous categories. Analyze environmental risks associated with improper disposal of e-waste and propose an eco-friendly e-waste management strategy for the institution.

3. Analysis and Optimization of Green Business Processes: Many organizational processes involve excessive paper usage, redundant steps, and inefficient resource utilization. Select a business process such as student admission, library management, or billing system. Study the existing workflow and identify areas where resources are wasted. Redesign the workflow using green business process management principles to minimize paper usage, reduce energy consumption, and improve efficiency. Represent the optimized workflow using a process modeling diagram.

4. Design of Green Enterprise Architecture for IT Infrastructure: Organizations rely heavily on IT infrastructure consisting of servers, storage systems, and networking equipment that consume significant energy. Study a typical enterprise IT infrastructure and identify components that contribute to high energy consumption. Propose a green enterprise architecture by incorporating technologies such as virtualization, cloud computing, and energy-efficient networking devices. Illustrate the improved architecture using a suitable system diagram and explain how it reduces environmental impact.

5. Data Center Energy Efficiency Analysis using PUE: Data centers require large amounts of electricity for servers, cooling systems, and networking devices. Study the concept of Power Usage Effectiveness (PUE) as a metric for measuring energy efficiency. Analyze a hypothetical or sample data center configuration by considering power consumed by IT equipment and supporting infrastructure. Calculate the PUE value and evaluate whether the data center is energy efficient. Suggest possible improvements such as advanced cooling techniques or renewable energy integration.

6. Implementation of Virtualization for Energy-Efficient Computing: Running multiple servers on separate physical machines results in inefficient resource utilization and increased energy consumption. Implement virtualization by installing multiple virtual machines on a single host system. Monitor CPU, memory, and resource usage before and after virtualization. Analyze how server consolidation reduces hardware requirements and contributes to energy-efficient computing.

7. Machine Learning-Based Prediction of Energy Consumption: Energy consumption in computing systems often varies depending on workload and system activity. Using a dataset containing historical energy consumption values, implement a machine learning model to predict future energy usage. Train and test the model using suitable regression techniques and evaluate prediction accuracy. Analyze how predictive models can help organizations optimize energy usage and reduce operational costs.

8. Development of Energy Monitoring and Visualization Dashboard: Monitoring energy consumption helps identify inefficient systems and reduce unnecessary power usage. Develop a dashboard that reads energy consumption data from a dataset and visualizes patterns using charts and

graphs. Identify peak consumption periods and propose recommendations to improve energy efficiency in computing infrastructure. https://archive.ics.uci.edu/dataset/235/individual+household+ele

Part-B: Tools Used for Group B Assignments

- Internet Research Databases
- IEEE / ACM Research Papers
- Microsoft Excel or Google Sheets
- Word Processing and Presentation Tools
- Python (optional for analysis)

Part B: List of Assignments

1. Computing devices contribute significantly to energy consumption and carbon emissions. Prepare a report analyzing the carbon footprint generated by computing devices used in educational institutions or offices. Study existing sustainable practices such as power management, device recycling, and reduced paper usage. Suggest practical steps for promoting green computing awareness within the organization.
2. Technology companies are increasingly adopting sustainable supply chain practices to reduce environmental impact. Select a well-known IT company and study its supply chain management strategies. Analyze how eco-friendly packaging, efficient logistics, and responsible sourcing contribute to sustainability. Present your findings in the form of a structured analysis.
3. Traditional data centers often suffer from low resource utilization and high operational costs. Study the concept of cloud computing and analyze how cloud infrastructure improves resource utilization and energy efficiency. Compare traditional server infrastructure with cloud-based solutions and prepare a comparative analysis highlighting environmental benefits.
4. Cooling systems consume a large portion of energy in data centers. Research AI-based cooling technologies used in modern data centers. Analyze how predictive algorithms monitor temperature patterns and automatically adjust cooling levels to maintain optimal conditions while minimizing energy consumption.

Part C: Mini-Project Any one (In Team of 3-4 Students)
Tools Used for Mini Projects

- Python Programming Language
- Jupyter Notebook
- Streamlit / Power BI for dashboards
- irtualBox / VMware
- Excel / MATLAB
- IoT Simulation Tools (optional)

Part C: Suggested List of Mini Project

1. Educational institutions operate numerous computing devices such as desktops, servers, printers, and networking equipment. Conduct an energy audit of computer laboratories by collecting data on device usage patterns and electricity consumption. Analyze inefficiencies and recommend strategies such as power management policies, device upgrades, and reduced paper usage to promote green computing.

2. Design a complete green IT infrastructure for a medium-sized organization. The proposed architecture should incorporate virtualization, cloud computing, and energy-efficient networking equipment. The design should aim to reduce power consumption, improve resource utilization, and support sustainable IT operations.Develop a simulation model to study how workload distribution and server consolidation affect energy consumption in a data center. The simulation should demonstrate how virtualization and efficient resource allocation reduce overall energy usage.

3. Develop a software system that collects energy usage data from computing devices and predicts future consumption patterns using machine learning algorithms. The system should visualize energy usage trends and provide recommendations for optimizing energy efficiency.

Learning Resources

Text Books

1. Hwaiyu Geng, Data Center Handbook, 2nd Edition, John Wiley & Sons, 2014. ISBN: 978-1118436639

2. San Murugesan, G. R. Gangadharan (Editors), Harnessing Green IT: Principles and Practices, 1st Edition, John Wiley & Sons / IEEE Press, 2012. ISBN: 978-1119970057

3. Bhuvan Unhelkar, Green IT Strategies and Applications: Using Environmental Intelligence, 1st Edition, CRC Press / Taylor & Francis, 2011. ISBN: 978-1439815977

Reference Books

1. Carl Spathis, Kevin Curran, Green Computing: Tools and Techniques for Saving Energy, Money and Resources, 1st Edition, CRC Press, 2012. ISBN: 978-1439879283

2. Jason Harris, Green Computing and Green IT: Best Practices on Regulations and Industry, 1st Edition, CreateSpace Independent Publishing, 2008. ISBN: 978-1440439612

3. Bud E. Smith, Green Data Centers: Steps for the Journey, 1st Edition, Prentice Hall / IBM Press, 2009. ISBN: 978-0137043750

4. Jorge Marx Gómez, Bernhard Niemann, Richard Thomas, Green IT: Technologies and Applications, 1st Edition, Springer, 2012. ISBN: 978-3642221958

e-Books: IEEE Xplore Digital Library – Green Computing Research Papers

1. Association for Computing Machinery Digital Library – Sustainable Computing Research

2. Sustainable Computing

3. Sustainability, Green IT and Education Strategies in the Twenty-first Century

MOOC Courses

1. Green Computing – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc20_cs79/preview

2. Cloud Computing – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs18/preview

3. Internet of Things – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs06/preview

4. Sustainable Engineering – IIT Madras https://onlinecourses.nptel.ac.in/noc22_ce95/preview

Savitribai Phule Pune University **Third Year - Computer Engineering (2024 Pattern)**		
VSE372COM - Solar Technology and Maintenance		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 50 Marks

Prerequisites: Basic knowledge of Physics (especially topics like electricity, magnetism semiconductors,light/energy concepts, Basic Electrical Engineering or Basic Electronics, Engineering Mechanics. Heat and
energy concepts

Course Objectives: The course aims to:

1. Apply Safely install, wire, and commission basic solar PV systems while measuring key performance parameters.
2. Analyze Break down the impact of environmental and operational factors on solar system efficiency and diagnose common faults.
3. Evaluate Judge the effectiveness of maintenance and troubleshooting procedures for solar PV components and systems.
4. Create Develop simple practical solutions or documentation for improving solar system performance in mini-projects.

Course Outcomes: Upon successful completion of this course, students will be able to:

- **CO1:** Apply safe installation, wiring, commissioning, and performance measurement of basic solar PV systems.
- **CO2:** Analyze the impact of environmental/operational factors on solar PV efficiency and diagnose common faults.
- **CO3:** Evaluate the effectiveness of maintenance and troubleshooting procedures for solar PV components and systems.
- **CO4:** Create simple practical solutions or documentation for improving solar system performance via mini-projects.

Practical Assignments

1. Experiment no.1, 2 and 10 are compulsory.
2. Perform any 2 Experiments from 3 to 5
3. Perform any 3 Experiments from 6 to 9s

1. Measurement of solar irradiance using pyrometer/lux meter at different times/angles. Real-World Assignment: Survey irradiance on your college rooftop for one day. Calculate daily energy generation for a 100W panel and suggest best installation time/angle for maximum output.
2. Plot I-V and P-V characteristics of solar PV module under varying light & temperature. Real-World Assignment: Simulate cloudy/rainy day conditions. Calculate module efficiency and estimate annual energy loss in Pune climate.
3. Survey and Comparative Analysis of Solar PV Installation Systems: Grid-Tied, Hybrid, and Off-Grid Configurations. Real-World Assignment: Survey 2–3 real solar installations (e.g., college rooftop, nearby home/business, or online/virtual

4. Series and parallel connection of PV modules, observe mismatch issues. Real-World Assignment: Design a small array for 12V/24V system (e.g., for laptop charging or lab fan). Calculate total power and suggest fuse/ diode protection for mismatch in a multi-panel rooftop installation.
5. Installation and wiring of standalone solar PV system (PV $\rightarrow$ Charge controller $\rightarrow$ Battery $\rightarrow$ Load/Inverter) Real-World Assignment: Prepare a complete wiring diagram and BOM for a 100W system to power a college water cooler or hostel room. Include safety earthing and cable sizing as per real IEC standards.
6. Preventive maintenance: Cleaning, visual inspection, corrosion/loose connection check. Real-World Assignment: Inspect any existing solar panel in college/hostel. Prepare a 6-month maintenance schedule with cost estimation (dust cleaning, tightening)
7. Grid-Related Maintenance Checks for Grid-Tied Solar PV Systems: Inverter Health, Performance Monitoring, and Fault Diagnosis. Real-World Assignment: Survey a real grid-tied installation, Prepare a maintenance schedule:Monthly inverter check, quarterly visual, annual professional inspection.
8. Mounting structure assembly: Rooftop/ground mount, tilt adjustment, stability check Real-World Assignment: Design a simple mounting frame for windy Pune conditions. Calculate wind load and suggest material/cost for a 5kW residential installation.
9. IoT-Based Real-Time Solar PV System Monitoring and Performance Dashboard.
10. Industrial Visit to Solar Energy Facility in Pune Region: Hands-On Learning of Solar PV System Operations and Maintenance

Test Books:

1. S.P. Sukhatme, Solar Energy
2. C.S. Solanki, Solar Photovoltaics
3. D.P. Kothari et al., Renewable Energy Sources
4. G.D. Rai, Non-Conventional Energy Sources
5. H.P. Garg, Solar Energy Utilization

Reference Books:

1. Solar Photovoltaic Technology and Systems: A Manual for Technicians, Trainers and EngineersAuthor: Chetan Singh Solanki.
2. Solar PV System: Design, Installation, Operation and MaintenanceAuthors: L. Ashok Kumar and K. Mohana Sundaram.
3. Solar Engineering of Thermal Processes, Photovoltaics and Wind (5th Edition) Authors: John A. Duffie, William A. Beckman (updated with Nathan Blair).
4. Principles of Solar Engineering (3rd Edition) Authors: D. Yogi Goswami, Frank Kreith, Jan F. Kreider

NPTEL Course:

1. Solar Photovoltaics: Fundamentals, Technology and Applications: https://onlinecourses.nptel.ac.in/noc
2. SkillCat or Other Free Solar Training (Installation Focus). https://www.skillcatapp.com/solar-installation-training

Savitribai Phule Pune University, Pune

Maharashtra, India



Task Force for Curriculum Design and Development

Programme Coordinator

Dr. Dipak Patil - Member, Board of Studies - Computer Engineering

Core Committee Members

Dr. Neeta Deshpande	Dr. Sharmila Wagh
Dr. Kamini Shirsath	Dr. Uday Patkar
Dr. P.M. Yawalkar	

Team Members for Course Design

Computer Networks	
Dr. S.R. Lahane	RH Sapat College of Engg, Management Studies and Research
Dr. Ranjana Dahake	MET's Institute of Engineering, Nasik
Dr. Salunke Mangesh	Marathwada Mitra Mandal's Institute of Technology
Dr. Megha Kadam	Marathwada Mitra Mandal's Institute of Technology
Mrs. Pallavi Rahinj	Rajiv Gandhi College of Engineering,Karjule Harya
Mr. Prasad Bhosle	Trinity Academy of Engineering, Pune

Artificial Intelligence	
Dr. Neeta Deshpande	RH Sapat College of Engg, Management Studies and Research
Dr. Vaishali Tidke	MVPS's KBTCOE Nashik
Dr. Chandrakant D. Kokane	Vishwakarma Institute of Technology, Pune
Dr. Vaibhav Dabhade	MET's Institute of Engineering
Dr. Suvarna Patil	D Y Patil International University, Akurdi, Pune
Mrs. Megha Amey Patil	RH Sapat College of Engg, Management Studies and Research

Theory of Computation	
Dr. Shilpa Khedkar	MES Wadia College of Engineering Pune.
Dr. P.M.Yawalkar (MET)	MET's Institute of Engineering, Nashik
Dr. Shwetal K. Patil	Marathwada Mitramandal's Institute of Technology, Lohgaon
Ms. Varsha Mhaske	College of Engineering Malegaon
Mrs.Pradnya Kothawade	Genba Sopanrao Moze College of Engineering

Data Warehousing and Data Mining	
Dr. Dipak V. Patil	RH Sapat College of Engg, Management Studies and Research
Dr. Sulochana Sonkamble	JSPM Narhe Technical Campus
Dr. Archana Pritam Kale	Modern Education Society's College of Engineering, Pune
Mrs. Pradnya Bachhav	GGS College of Engineering & Research Centre Nasik
Ms. Sunita Borse	MET's Institute of Engineering, Nashik

Cloud Computing	
Dr. Kamini Shirsath	Sandip Institute of Engineering and Management Nasik
Dr. Renuka Gound	Nutan Maharashtra Institute of Engineering and Technology, Pune
Dr. Madhuri Shinde	MET's Institute of Engineering, Nashik
Mrs. Nanda Kulkarni	Siddhant College of Engineering Pune
Mr. Bhushan Pawar	Software Technology Architect, Congnizant

Mobile Computing	
Dr. Nikita Kulkarni	K J College of Engineering and Management Research Pune
Mrs.. Shubhangi S. Satav	Bharati Vidyapeeth's College of Engineering Lavale Pune.
Mrs. Mayuri Vengurlekar	KJ College of Engineering and Management Research
Mr. Sujit B. Bhonkar	Bharati Vidyapeeth's College of Engineering Lavale Pune

Embedded Systems	
Dr. Archana Banait	MET's Institute of Engineering,Nasik
Dr. Prashant Salunkhe	GOGATE Electosystem Pvt.Ltd,Nasik
Dr Sujeet More	Trinity college of Engineering and Research ,Pune
Dr. Supriya sable	MIT ,Academy of Engineering Alandi,Pune
Mrs Amrita Pundlik	GES's R H Sapat COEMSR , Nashik

Software Engineering and Modelling	
Dr. Uday Chandrakant Patkar	Bharati Vidyapeeth's College of Engineering Lavale Pune.
Dr. Balasaheb S. Tarle	MVPS's KBT College of Engineering,Nasik
Dr. Rahul Gaikwad	Amrutvahini College of Engineering, Sangamner
Mr.Sagar Dhanake	Dr. D Y Patil College of Engineering and Innovation, Varale, Talegaon
Ms. Charushila Patil	Guru Gobind Singh College of Engineering & Research Centre

Quantum Computing	
Ms. Vrushali Nikam	R.H.Sapat College of Engineering,Nashik
Mr. Nilesh Pinjarkar	JSPM's Rajarshi Shahu College of Engineering
Mrs. Kumkum Bala	Bharati Vidyapeeth's College of Engineering Lavale Pune
Mrs Parinita Chate	Bharati Vidyapeeth's College of Engineering Pune

Distributed Systems	
Dr. P.M. Yawalakar	MET's Institute of Engineering, Nashik
Dr. Ganesh Puri	Amrutvahini College of Engineering
Dr. V. S. Pawar	Karmaveer Adv. Baburao Ganpatrao Thakare College of Engineering
Dr. Vijay Rathod	VIT Pune
Mr. Kiran Kulkarni	MET's Institute of Engineering, Nashik

System Programming	
Dr. Shraddha Konde	MES Wadia College of Engineering, Pune
Ms. Charushila Patil	GGS College of Engineering and Research Center, Nashik
Dr. Vaishali Khandave	MET's Institute of Engineering, Bhujbal Knowledge City, Nashik
Mrs. Abha Pathak	DYPCOEI
Mrs. Pallavi Yevale	I2IT, Pune
Mrs. A.S.Deokar	AISSMS, Pune

Information Retrieval	
Dr. Jayashri Shinde	MMIT Lohegaon
Dr. Bajirao Shirole	Loknete Gopinathji Munde Institute Engineering Education and Research
Ms. Pranjali Deshmukh	MMIT Lohegaon
Ms. Shital S. Patil	Keystone School of Engineering
Dr. Snehal A Karad	Marathwada Mitramandal Institute of Technology Pune

Data Visualization and Analytics	
Dr. Shalaka Deore	Modern Education Society's College of Engineering
Mrs. Mangal Kale	Dr. Vittalrao Vikhe Patil CoE Ahilyanagar
Ms. Sunita Borse	MET's Bhujbal Knowledge City, Nasik
Dr. Sonali B. Shirke	Keystone School of Engineering

Cryptography and Network Security	
Dr. Sonali Shirke	Keystone School of Engg
Dr. Mangesh D. Salunke	Marathwada Mitramandal Institute of Technology Pune
Dr. Hrushikesh J. Joshi	Vishwakarma Institute of Technology, Pune
Dr.Sarika S. Jadhav	Padmabhushan Vasantdada Patil Institute of Technology, Pune

Advanced Databases	
Dr. Sharmila Wagh	M.E.S. Wadia College of Engineering, Pune
Dr Sharau Lokhande	Army Institute of Technology ,Pune
Prof Ratnakar Jagale	RH Sapat College of Engineering, Management Studies And Research
Prof Sagar Shinde	MES Wadia College of Engineering Pune
Dr. Archana Vaidya	GES's RH Sapat College of Engineering and Management Studies, Nasik

Green Computing	
Dr. Archana Vaidya	GES's R H Sapat COEMSR , Nashik
Dr. D. P. Gaikwad	AISSMCOE, Pune
Dr. Brijendra Gupta	Siddhant COE, Pune
Dr. Yogita D. Sinker	VP's KB IoT, Baramati
Dr. Renuka Kajale	PCET's Nutan Maharashtra Institute of Engineering and Technology

Chairman

Dr. Nilesh Uke - Board of Studies Computer Engineering

Savitribai Phule Pune University, Pune

Associate Dean

Dr. Gajanan Kharate - Associate Dean – Science and Technology
Savitribai Phule Pune University, Pune

Dean

Dr. Raosaheb Latpate - Dean – Science and Technology
Savitribai Phule Pune University, Pune

***