Cummins College of Engineering for Women

(An autonomous institute affiliated to Savitribai Phule Pune University)
Karve Nagar, Pune - 411 052.

MAHARSHI KARVE STREE SHIKSHAN SANSTHA'S
CUMMINS COLLEGE OF ENGINEERING FOR WOMEN

Vision

To be globally renowned engineering institute for imparting holistic education and developing professional women leaders in engineering and technology

Structure and Syllabus

of

T. Y. BTech (Information Technology)

2023 Pattern [R0]

List of Abbreviations

Abbreviation	Title
PC	Programme Core Course
BSC	Basic Science Course
ESC	Engineering Science Course
PE	Programme Elective Course
OE	Open Elective
VSEC	Vocational and Skill Enhancement Course
CC	Co-curricular Courses / Liberal Learning Course
IKS	Indian Knowledge System
VEC	Value Education Course
RM	Research Methodology
INTR	Internship
PROJ	Project
CEP	Community Engagement Project
Mm	Multidisciplinary Minor
AEC	Ability Enhancement course

MKSSS's Cummins College of Engineering for Women, Pune

(An Empowered Autonomous Institute, affiliated to Savitribai Phule Pune University)

Curriculum for UG Degree Course in BTech. Information Technology

(Academic Year: 2025-26 Onwards)

Third Year | Semester-VI

Sr. No.	Course Code	Course Title	Teaching Scheme Hours /Week			Credits	Examination Scheme			Total Marks
			L	T	P		ISE	ESE	Pr/Or	
1	23PCIT601	Software Engineering	3	0	0	3	50	50	0	100
2	23PCIT602	Information and Cyber Security	3	0	0	3	50	50	0	100
3	23PCIT603	Data Science	3	0	0	3	50	50	0	100
4	23PCIT604	Cloud Computing and DevOps	3	0	0	3	50	50	0	100
5	23PEIT601	Programme Elective-II	3	0	0	3	50	50	0	100
6	23MmIT601	Multidisciplinary Minor Course 3	2	0	0	2	25	25	0	50
7	23PCIT602L	Information and Cyber Security Laboratory	0	0	2	1	25	0	25	50
8	23PCIT603L	Data Science Laboratory	0	0	2	1	25	0	25	50
9	23PEIT601L	Programme Elective-II Laboratory	0	0	2	1	25	0	25	50
10	23VSEC601	Programming Skills in DevOps Laboratory	0	0	4	2	50	0	0	50
		Total =	**17**	**00**	**10**	**22**	**400**	**275**	**75**	**750**

L=Lecture, T=Tutorial, P= Practical, Cr= Credits, ISE =In Semester Evaluation, ESE =End Semester Examination, Pr/Or = Practical/Oral

23PEIT601: Program Elective-II

A. Deep Learning
B. Animation and Gamification
C. Blockchain

23PEIT601L: Program Elective-II Laboratory

A. Deep Learning Laboratory
B. Animation and Gamification Laboratory
C. Blockchain Laboratory

23MmIT601: Multidisciplinary Minor Course 3

Green Data Center Design and Management

23PCIT601 Software Engineering

Teaching Scheme:	**Examination Scheme:**
Lectures:3 hours/week	**In-Semester:** 50 marks
	End-Semester: 50 marks
	Credits: 3

Prerequisites: Programming fundamentals, Databases

Course Objectives:

Familiarize students with

1. Linear nature of software complexity in various application domains, disciplined way of software development and software lifecycle process models.
2. Iterative concepts and principles of software design and architecture.
3. Basics of software testing through real life projects
4. Recent trends in software engineering.

Course Outcomes:

Students should be able to

1. Apply process models and practices for diverse software domains.
2. Analyze software requirements to derive suitable design solutions for real-life systems.
3. Evaluate software quality using systematic testing, debugging and maintenance practices.
4. Make use of modern tools for software project management.

Unit I Introduction to Software Engineering

The nature of software, Software application domains, software process, Software engineering practice, Guiding principles, Software myths, Software process models, Prescriptive models, Specialized models, Software build process, Process assessment and improvement, documentation standards

Unit II Software Requirement Analysis

Requirements Engineering: Elicitation, Specification, Validation, Negotiation, Prioritizing requirements - kano diagram

Requirements analysis and modeling: Objectives and philosophy, Scenario-based modelling, Use case model, Domain-driven design, Data and control flow model, Behavioural modelling

Unit III Software Design

Design concepts: Abstraction, Architecture, Design patterns, Separation of concerns, Modularity, Information hiding, Functional independence, Design classes.

Architectural design: Software architecture, Architectural styles and patterns.

Component level design: Class based components, Component level design.

User interface design: Golden rules, UI Analysis, Narratives, Personas and scenarios.

Unit IV Software Testing

Software testing basics, Unit testing and integration testing, White box and Black box testing, Regression testing, Peer testing, Performance testing, Security testing, Debugging techniques, Software maintenance, Automated testing frameworks,

Unit V Software Project Management

Project metrics and measurement, Planning, Scope management, Metrics for size estimation, Project cost estimation, Work Breakdown Structure, Effort and schedule estimation, Project scheduling Principles, Scheduling techniques, Project tracking and control, Timeline charts, Earned value analysis, Gantt Charts

Unit VI Recent trends in Software Engineering

Computer-aided software engineering (CASE), Software configuration management tools,
Risk management, Agile development process, SCRUM, Introduction to AI and Prompt
Engineering, Sustainability in software engineering
Project management trends: ERP, SAP, Global software development,

Textbooks

1. R. Roger S. Pressman, "Software Engineering: A practitioner's approach", McGraw Hill, 9th Edition, 2023.
2. Ian Sommerville, "Software Engineering", Addison-Wesley, 10th Edition, 2021.

Reference Books

1. Karl E. Wiegers, Joy Beatty, "Software Requirements", Microsoft Press, 3rd Edition, 2014.
2. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Addison-Wesley, 4th edition, 2022.
3. Dorothy Graham, Rex Black, Erik van Veenendaal, "Foundations of Software Testing", Cengage Learning, 5th edition, 2024.
4. Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", McGraw Hill, 6th edition, 2017.
5. James Shore & Shane Warden, "The Art of Agile Development", O'Reilly Media, 2nd Edition, 2021.

23PCIT602 Information and Cyber Security

Teaching Scheme:	**Examination Scheme:**
Lectures : 3 hours / week	**In-Semester:** 50 marks
	End-Semester: 50 marks
	Credits:3

Prerequisites: Computer Networks, Operating Systems

Course Objectives:

Familiarize students with

1. Fundamental principles, models, and goals of information security.
2. Cryptographic algorithms and key management mechanisms
3. Threats, vulnerabilities, malware behaviors, and network-level security challenges.
4. Security issues in applications, and web systems and understand essential mitigation practices.

Course Outcomes:

Students should be able to

1. Apply information security principles and models to protect systems and data.
2. Apply cryptographic algorithms and key management techniques to ensure secure communication.
3. Analyze threats, vulnerabilities, malware, and network attacks to identify appropriate security controls.
4. Analyze security weaknesses in applications and web systems to suggest effective countermeasures.

Unit I Foundations of Information Security and Cryptography

Overview of Information Security – CIA triad, threats, vulnerabilities, and risks, data privacy.
Authentication mechanisms – passwords, MFA, biometrics, and SSO.
Classical cryptography – substitution and transposition ciphers.
Cryptanalysis and countermeasures against common attacks.
Introduction to operating system security and access control models – DAC, MAC, RBAC.

Unit II Modern Cryptographic Techniques

Modern Cryptographic Algorithms: Symmetric Encryption: DES, AES, Asymmetric Encryption: RSA, Diffie-Hellman, Cryptographic Hash Functions: MD5, SHA-1, Message Authentication Codes (MAC)

Unit III Key Management and Distribution

Digital signatures, public-key certificates, X.509, CA hierarchies, and Kerberos.

Unit IV Malware Fundamentals and Analysis

Types of malware : viruses, worms, trojans, ransomware, and zero-day attacks.
Malware delivery mechanisms : phishing, drive-by downloads, and exploit kits.
Cyber Kill Chain : stages of attack and defense strategies.
Malware analysis : static and dynamic techniques, detection, and prevention methods.

Unit V Network Security and Secure Communications

Network security fundamentals : OSI/TCP-IP models, common network attacks, firewalls, and intrusion detection/prevention systems.
Secure communication protocols : SSL/TLS, VPNs, IPSec, and secure email (S/MIME, PGP).
Security design principles : least privilege, defense in depth, fail-safe defaults, and secure system architecture.

Unit VI Application and Web Security

Application and API Security: Secure coding practices, input validation, sanitization, session management, API threat mitigation, secure API design, and token/key protection.
Web Security: HTTP/HTTPS security concepts, secure headers, and common web vulnerabilities.

Textbooks

1. W. Stallings, "Cryptography and Network Security: Principles and Practice", 9th ed. Pearson, 2023.
2. J. Andress, "The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice", 3rd ed. Syngress, 2022.

Reference books

1. R. Oppliger, "Cryptography 101: From Theory to Practice", Artech House, 2021.
2. D. R. Stinson and M. B. Paterson, "Cryptography: Theory and Practice", 4th ed., CRC Press, 2018.
3. R. Panko, "Corporate Computer and Network Security", 2nd ed., Pearson, 2019.
4. J. M. Stewart, M. Chapple, and D. Gibson, "(ISC)^2 CISSP Certified Information Systems Security Professional Official Study Guide", 9th ed, Wiley, 2021.
5. S. Harris and F. Maymi, "CISSP All-in-One Exam Guide", 8th ed., McGraw-Hill Education, 2021.
6. D. Tevault, "Mastering Linux Security and Hardening", 3rd ed., Packt, 2021.
7. T. Kalsi, "Practical Linux Security Cookbook", Packt, 2019.
8. M. E. Russinovich, D. A. Solomon, and A. Ionescu, "Windows Internals", 7th ed., Redmond, WA: Microsoft Press, 2017.
9. G. Johansen, "Digital Forensics and Incident Response", 2nd ed., Packt, 2020.
10. T. Jaeger, "Operating System Security", Boston, MA: Addison-Wesley, 2012.

23PCIT603 Data Science

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester:50 Marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Statistics and linear algebra

Course Objectives:
Familiarize students with
1. Fundamentals of data science
2. Techniques for data wrangling, exploration, visualization
3. Analysis of data for prediction and forecasting
4. Handling big data and data processing frameworks

Course Outcomes:
Students will be able to:
1. Explain the fundamental concepts, lifecycle, applications, and ethical considerations of Data Science and its data types
2. Apply techniques for data collection, cleaning, transformation, integration, and exploratory analysis to prepare datasets for meaningful insights
3. Analyze data using statistical methods and modern tools to visualise findings effectively through dashboards and storytelling
4. Solve real-world problems efficiently with big data, AutoML, and emerging trends in Data Science

Unit I Fundamentals of Data Science

Types of data: Structured, Semi-structured, Unstructured, Scales of measurement: Nominal, Ordinal, Interval, Ratio, Data formats, Data quality dimensions, DIKW pyramid, Introduction to Data Science, The Data Science Process/Lifecycle (e.g., CRISP-DM, Team Data Science Process (TDSP), SEMMA Methodology), Applications of Data Science, Ethical and Privacy Issues in Data Science.

Unit II Statistical Foundations of Data Science

Probability Distributions: Normal, Binomial, Poisson, etc., Sampling and Sampling Distributions, Hypothesis Testing (t-test, chi-square test) and Confidence Intervals, Regression Analysis: Simple and Multiple Linear Regression, Statistical Inference and P-values, Basics of time series data and forecasting, Moving averages and exponential smoothing, ARIMA and SARIMA models

Unit III Data Collection and Data Wrangling

ETL, Data Collection Techniques: Web Scraping, APIs, Sensor Data, and Databases, Data Cleaning: Handling Missing Data, Outliers: (Z-Score Method, Interquartile Range), and Duplicates, Data Transformation: normalization, standardization, and encoding, Data Integration and Merging Multiple Datasets, Exploratory Data Analysis (EDA): summary statistics, Univariate, bivariate, multivariate analysis.

Unit IV Data Exploration and Visualization

Descriptive Statistics: Measures of Central Tendency and Dispersion, Data Exploration Techniques for Numerical and Categorical Variables, Visualization Principles and Techniques, Data Visualization Tools, Advanced Visualization: Heatmaps, Pair Plots, Histograms, and Geographic Maps, Dashboard Basics with Tableau or Power BI, Data storytelling for effective communication and presentation

Unit V Recent trends in Data Science

Big Data, AutoML: AutoML Workflow, AutoML Libraries, Benefits and Limitations of AutoML, AutoML vs Traditional Machine Learning, data preprocessing for AutoML, data normalization and standardization for AutoML, Recent Trends: Federated Learning, Data-Centric AI, Data Democratization, Quantum Computing in Data Science

Textbooks

1. Vijay Kotu, Bala Deshpande, "Data Science Concepts and Practice", 2nd Edition, Morgan Kaufmann, ISBN: 978-0-12-814761-0., 2019
2. Gilbert Strang, "Introduction to linear algebra", 6th Edition, Wellesley-Cambridge Press, ISBN: 978-17331466-7-8, 2023

Reference Books

1. Ofer Mendelevitch, Casey Stella, Doug Eadline, "Practical Data Science with Hadoop and Spark: Designing and Building Effective Analytics at Scale, 1st Edition, Addison-Wesley, ISBN: 978-0134024141, 2017
2. Jeffrey Saltz and Jeffrey Stanton, "An Introduction to Data Science", 1st Edition, Sage Publications, ISBN: 978-1506377537, September 2017
3. Wes McKinney, "Python for Data Analysis", 3rd Edition, O'Reilly Publication, ISBN: 9781098104030, 2022
4. Suresh Kumar Mukhiya, Usman Ahmed, "Hands-On Exploratory Data Analysis with Python" 1st Edition, 2020, Packt Publish, ISBN 978-1-80323-110-5.
5. Jake Vanderplas, "Python Data Science Handbook", O'Reilly Publications, ISBN: 9781491912126, 2016
6. Davy Cielen, Arno D.B. Meysman, Mohamed Ali, "Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools", 1st Edition, Dreamtech Press, ISBN 978-1-63343-003-7.
7. Montgomery, Douglas C.; Runger, George C., "Applied Statistics and Probability for Engineers" 2nd Edition, Wiley, ISBN 10: 0471170275, 1998

23PCIT604 Cloud Computing and DevOps

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 hours/week	**In-Semester:** 50 marks
	End-Semester: 50 marks
	Credits: 3

Course Objectives:

Familiarize students with

1. Principles of cloud computing
2. Principles and practices of DevOps.
3. Automation tools, configuration management, CI/CD pipeline.
4. Dockers and Kubernetes
5. Trends in DevOps.

Course Outcomes:

Students will be able to:

1. Explain core concepts and fundamentals of cloud computing.
2. Describe principles, practices, and necessity of DevOps in software engineering.
3. Apply building, testing strategies and microservices architecture for implementing CI/CD.
4. Use various strategies for software deployment, containers, and orchestration.
5. Explain emerging trends in DevOps.

Unit I Introduction to Cloud Computing

Overview of distributed systems and cloud computing, virtualization: hypervisors, virtual machines, virtual clusters, cloud service models: IaaS, PaaS, SaaS, public cloud platforms: AWS, Azure, GCP.

Unit II DevOps Fundamentals and CI/CD

DevOps principles, culture, and lifecycle, automation in DevOps, continuous integration (CI) and continuous deployment (CD) pipelines, infrastructure as code (IaC) tools: Terraform, Chef, Puppet

Monolithic vs microservices architecture

Unit III Build and Test Automation

Build automation tools: Make, Ant, Maven, importance of automated builds, testing strategies: unit tests, integration tests, regression tests, smoke tests, packaging applications for deployment.

Unit IV Deployment Strategies and Release Management

Deployment concepts and reasons, deployment strategies: blue/green, canary, A/B testing, feature toggling, rollback, Managing multiple versions, logical consistencies, and race conditions.

Unit V Containerization, Orchestration, and Emerging Trends

Containers vs VMs, docker architecture, images, and registries, orchestration using Kubernetes: pods, services, deployments.

Emerging DevOps trends: GitOps, MLOps, AIOps, DataOps, DevSecOps.

Textbooks

1. Ekamber K Singirikonda, DevOps Automation Cookbook, BPB Publications, 1st Edition, 2024.
2. Sam Newman, Building Microservices: Designing Fine-grained Systems, O'Reilly, 2nd Edition 2021.

Reference Books

1. Jack J. Dongarra, Kai Hwang, Geoffrey C. Fox, "Distributed and Cloud Computing: From Parallel Processing to the Internet of Things", Elsevier, 1st Edition.
2. Mikael Krief, Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins, 2nd Edition, March 2022
3. Sanjeev Sharma, The DevOps Adoption Playbook, A guide to adopting DevOps in a Multi-Speed IT Enterprise. Wiley, IBM Press, April 2017.
4. Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann, "Introducing MLOps", O'Reilly Nov 2020
5. Billy Yuen, Jesse Suen, Alex Matyushentsev, Todd Ekenstam, "GitOps and Kubernetes", O'Reilly, April 2021
6. Gene Kim, Jez Humble, Patrick Debois, John Willis, and Nicole Forsgren, The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, 2nd Edition, October 2016
7. Len Bass, Ingo Weber, Liming Zhu, DevOps A Software Architect's Perspective, Pearson, May 2015.
8. Jennifer Davis and Katherine Daniels, "Effective DevOps", O'Reilly, 1st Edition
9. Jez Humble, David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation"
10. Deepak Gaikwad, Viral Thakkar, "DevOps Tools from Practitioner's Viewpoint", Wiley, 1st Edition
11. Aditya Pratap Bhuyan," Docker Deep Dive: Learn, Build, and Scale"
12. Kelsey Hightower, Brendan Burns, and Joe Beda, "Kubernetes Up & Running: Dive into the Future of Infrastructure"

Other Resources:

1. https://azure.microsoft.com/en-in/solutions/devops/tutorial

23PEIT601A Deep Learning

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 hours/week	**In-semester:** 50 marks
	End-Semester: 50 marks
	Credits: 3

Prerequisites: Machine Learning

Course Objectives:

Familiarize students with:

1. Concepts of Deep Learning
2. Applications of Deep Learning
3. Types of Deep Learning techniques
4. Advanced trends in Deep Learning

Course Outcomes:

Students will be able to:

1. Explain basics of neural networks and deep learning
2. Use concepts of ANN and CNN to solve various problems.
3. Apply concepts of RNN and LSTM to sequential data
4. Identify importance of transformers in various deep learning applications

Unit I Artificial Neural Network

Artificial neural network, activation functions, backpropagation, chain rule, gradient descent, stochastic gradient descent

Unit II Convolutional Neural Network

Basic structure of CNN: filters, padding, strides, channels, pooling, fully connected layers, interleaving between layers, training a CNN and backpropagation, applications of CNN Case Study: Alex Net

Unit III Recurrent Neural Network and Long-Short Term Memory

Architecture of RNN time layered representation, different variations of RNN with missing inputs and outputs, bi-directional RNN, challenge of long-term dependencies, LSTM, applications of RNN

Case Study: Elmo

Unit IV Deep Reinforcement Learning

Multi-armed bandits, framework of RL, Q-Learning, deep learning models as function approximators

Case Study: AlphaGo

Unit V Advanced Topics in Deep Learning

Attention mechanisms, encoder-decoder sequence-to-sequence architecture, transformers architecture, LLM.

Case Study: BERT, GPT

Textbooks:

1. Charu C. Aggarwal, "Neural Networks and Deep Learning", Springer 2nd Edition - 2023.
2. Ethem Alpaydin, "Introduction to Machine Learning", PHI 4th Edition – 2020.

Reference Books:

1. Nikhil Buduma, "Fundamentals of Deep Learning – Designing Next Generation Machine Intelligence Algorithms", O'Reily – 1st Edition – 2017.
2. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly 1st Edition 2021.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press 2016.
4. Francois Chollet, "Deep Learning with Python", Manning 2nd edition 2021.

Other Resources

1. UCI Machine Learning Repository https://archive.ics.uci.edu
2. WEKA Collection of datasets https://waikato.github.io/weka-wiki/datasets/
3. Kaggle datasets https://www.kaggle.com/datasets

23PEIT601B Animation and Gamification

Teaching Scheme:
Lectures:3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Geometry and trigonometry, Vectors and Matrices

Course Objectives:

Familiarize students with

1. Concepts of Animation and Gamification
2. Basic primitives and objects in computer graphics
3. Various methods and techniques used in Animation and Gamification
4. Applications of computer graphics in animation and gaming

Course Outcomes:

Students should be able to

1. Apply animation principles, geometric transformations, and tools to create 2D and 3D animations for interactive platforms.
2. Apply the principles of animation and gamification to real life applications.
3. Design gamified systems using game mechanics, and animation techniques.
4. Explain applications of AR/VR, AI, and future trends.

Unit I Introduction to Computer Graphics

Basics of graphics systems, graphics primitives for drawing, 2D geometric transformations, 3D geometric transformations, polygon filling methods, clipping methods.

Unit II Principles of Animation

Basics of animation principles (e.g., squash and stretch, timing, easing), understanding gamification: definitions, concepts, and applications, differences and intersections between animation and gamification

The 12 principles of animation, keyframing, twining, and motion paths, 2d vs. 3d animation techniques, fundamentals of timing and spacing, interactive graphics and usage of the tools of computer graphics

Unit III Animation Techniques

Introduction to animation software, basics of character design and rigging, introduction to rendering and compositing

Unit IV Basics of Game Design and Gamification

Elements of game design: goals, rules, feedback, and engagement, motivational theories in gamification, levels, badges, points, and other game mechanics, ethics and best practices in gamification

Gaming platforms: graphics memory pipeline, block diagram of Nvidia workstation and i860 introduction to OpenGL

Unit V Integrating Animation into Gamification

Animation techniques for user engagement in gamified platforms, storytelling and narrative techniques in gamified contexts, designing animated feedback

Unit VI Advanced Applications and Trends in Animation and Gamification

AR/VR applications in animation and gamification, machine learning and AI in animation, case studies of successful gamified systems, future trends in gamified animation, metaverse

Textbooks

1. D. Hearn, M. Baker, "Computer Graphics with OpenGL", 4th Edition, Pearson Education, 2014, ISBN 978-93-325-1871-1
2. Andrew Hogue, Rick Parent, "Computer Animation Algorithms and Techniques", 3rd Edition , Morgan Kaufmann Publishers, 2012, ISBN 978-0124159730

Reference Books

1. G. Zichermann, C. Cunningham, "Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps", 1st Edition, O'Reilly Media, 2011, ISBN 978-1449397678.
2. J. Mathew, S. Nair, "Level Up Your Learning: The Power of Gamification", 1st Edition, Nova Science Publishers, 2024, ISBN 979-8891134096.
3. M. Dunsky, "C++ Game Animation Programming: Learn modern animation techniques with OpenGL and Vulkan" (2nd Edition), Packt Publishing, 2023, ISBN 978-1803246529.
4. M. Dunsky, "Mastering C++ Game Animation Programming", 1st Edition, Packt Publishing, 2025, ISBN 978-1835881927

23PEIT601C Blockchain Technology

Teaching Scheme:	**Examination Scheme:**
Lectures: 3 hours/week	**In-semester:** 50 marks
	End-Semester: 50 marks
	Credits: 3

Prerequisites: Operating systems, Computer Networks

Course Objectives:

Familiarize students with:

1. Blockchain fundamentals, cryptographic principles, and decentralized systems.
2. Consensus mechanisms and blockchain architectures used in real-world systems.
3. Smart contracts and decentralized applications.
4. Blockchain applications.

Course Outcomes:

Students will be able to:

1. Describe blockchain fundamentals, cryptographic methods, and decentralization principles.
2. Compare major consensus mechanisms and blockchain architecture.
3. Explain the principles and functioning of smart contracts and decentralized applications.
4. Evaluate blockchain applications and security mechanisms.

Unit I Blockchain Foundations

Characteristics: transparency, immutability, decentralization, centralized vs. decentralized systems, Distributed Ledger Technology (DLT), transactions and blocks, security, integrity, and privacy in decentralized systems.

Unit II Cryptographic Concepts for Blockchain

Structure of a block: hash, nonce, timestamp, Merkle tree. Hash functions: SHA-256, Keccak, symmetric and asymmetric cryptography, public/private key infrastructure, digital signatures and certificates

Unit III Consensus and Blockchain Architectures

Energy efficiency and scalability issues, bitcoin blockchain: UTXO model, Ethereum blockchain: account-based model, permissioned vs. permissionless blockchains, Layer-1 vs. Layer-2 solutions, Tokenization: concepts, enterprise applications, challenges

Unit IV Blockchain Platforms and Smart Contracts

Ethereum blockchain: features, components, ecosystem, Ethereum programming languages and bytecode, working of smart contracts, Smart contract design and deployment, introduction to Decentralized Applications (DApps), Hyperledger Fabric: architecture, principles, features differences between Ethereum and Hyperledger

Unit V Security in Blockchain Systems

Double-spending attacks, Sybil attacks, 51% attacks, privacy mechanisms: zk-SNARKs, ring signatures, mixers, Scalability solutions: sharding

Unit VI Applications and Future Directions

Applications: DeFi, supply chain, healthcare, e-voting, digital identity, Case studies: supply chain financing, health insurance, Regulatory and ethical issues: privacy, ransomware, money laundering, Legal aspects and GDPR, Blockchain governance and DAOs, Future trends: Web3.0, Metaverse, Blockchain + AI/IoT, Cryptocurrency

Textbooks:

1. Imran Bashir, Mastering Blockchain, 4th Edition, Packt Publishing, 2023.
2. Arvind Narayanan et al., Bitcoin and Cryptocurrency Technologies, Princeton University Press, 2016.

Reference Books:

1. Kumar Saurabh, Ashutosh Saxena, Blockchain Technology: Concepts and Applications, Wiley India, 2019.
2. Andreas M. Antonopoulos, Mastering Bitcoin, 2nd Edition, O'Reilly Media, 2017.
3. Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, 2015.
4. Roger Wattenhofer, The Science of the Blockchain, CreateSpace Publishing, 2016.

Other Resources

1. NPTEL: Blockchain and its Applications
2. NPTEL: Blockchain Architecture Design and Use Cases
3. Introduction to Blockchain Technology and Application

23MmIT601 Green Data Center Design and Management

Teaching Scheme:
Lectures:2 hours/week

Examination Scheme:
In-Semester: 25 marks
End-Semester: 25 marks
Credits: 2

Prerequisites: Green IT Infrastructure and Sustainability Metrics

Course Objectives:

Familiarize students with

1. Green data-center principles and sustainability concepts.
2. Modelling and analytical skills for layout and performance planning.
3. Automation and sustainability metrics for monitoring.
4. Design and management principles for sustainable solutions.

Course Outcomes:

Students should be able to

1. Explain the principles, standards, and architectural components of green data center design and management.
2. Analyze resource parameters for energy-efficient operation.
3. Evaluate the role of monitoring, automation, and sustainability metrics in optimizing data center performance.
4. Examine frameworks, standards, and compliance requirements for governing sustainable data centers.

Unit I Principles of Green Data Center Design

Design goals: scalability, efficiency, reliability, sustainability; data-center lifecycle: site selection, architecture, deployment, operation; Building–IT integration for sustainability; frameworks: The Green Grid, ASHRAE TC 9.9, Uptime Standards.

Unit II Power, Cooling and Space Planning

Power system design: redundancy (N, N+1, 2N), renewable integration, energy storage; cooling design: airflow modelling, containment, liquid/immersion cooling; CFD for airflow simulation; rack arrangement and spatial layout optimization.

Unit III Automation and Monitoring Systems

DCIM overview and components; Sensors, IoT, and real-time telemetry; AI/ML-based predictive maintenance and anomaly detection; Automation for workload placement and cooling control; carbon dashboards and sustainability reporting, carbon-neutral datacenters and circular IT ecosystems.

Unit IV Management, Security and Compliance

Physical, network, and cyber security in data center operations, secure automation: balancing security controls with energy efficiency, regulatory compliance frameworks, sustainable and compliant data centers in BFSI and Government sectors.

Textbooks

1. San Murugesan & G. R. Gangadharan, Harnessing Green IT: Principles and Practices, Wiley-IEEE Press, 2012.
2. Hwaiyu Geng, Data Center Handbook, Wiley, 2021

Reference Books

1. The Green Grid Association, Data Center Design and Operations Best Practices, 2024 Edition.
2. Vertiv (2025), Data Center Trends 2025: AI, Cooling, and Regulation.
3. Deloitte (2025), Sustainable Data centers and Generative AI Energy Insights.
4. MeitY (India, 2024), Green Data center Policy Guidelines.

23PCIT602L Information and Cyber Security Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical : 2 hours / week	**In-Semester:** 25 marks
	Practical: 25 marks
	Credits:1

Prerequisites: Computer Networks, Operating Systems

Course Objectives:

Familiarize students with

1. Fundamental concepts and techniques of information and cyber security.
2. Cryptographic algorithms and security mechanisms.
3. Analyzing system and network vulnerabilities using security tools.
4. Evaluating security solutions.

Course Outcomes:

Students should be able to

1. Apply fundamental security principles and techniques to ensure the CIA.
2. Implement security mechanisms for protecting data, applications, and networks.
3. Analyze programs/systems to identify vulnerabilities and risks.
4. Demonstrate testing, evaluation, and documentation of security solutions.

Suggested List of Laboratory Assignments

1. Implement the cipher algorithms for both encryption and decryption. Test the algorithms with various shift values and demonstrate how the shift impacts the encryption.
    a. Caesar
    b. Playfair
    c. Rail Fence
    d. Hill
2. Write a program to generate the public and private keys for the RSA algorithm. Implement the encryption and decryption processes to securely transmit a message.
3. Implement the Advanced Encryption Standard (AES) algorithm to perform both encryption and decryption operations on a user-provided input (string or file). Use any standard cryptographic library available in Java, Python, or C++ to ensure secure data handling.
4. Implement the MD5 hashing algorithm to compute the hash value (message digest) for various input strings. Use a built-in cryptographic library available in Java, Python, or C++ to ensure accurate and secure hash computation.
5. File Analysis : Explore the Portable Executable (PE) / Executable and Linkable Format (ELF) File Analysis file format using a tool like PEview or CFF Explorer or readelf. Analyze a given executable and identify sections, headers, and metadata.
6. XSS Vulnerability Exploitation: Use Burp Suite to find and exploit XSS vulnerabilities on a test website.

Additional List of Laboratory Assignments

1. File Analysis : Use PEStudio or Ghirda to analyze a sample PE or ELF file and identify its sections, imports, and exports.
2. Windows Firewall Configuration: Configure Windows Firewall to block or allow specific ports and applications. Test the effectiveness by checking port accessibility.
3. Linux Firewall Configuration: Configure a basic firewall using iptables on a Linux machine, defining rules for incoming and outgoing traffic.
4. SQL Injection Vulnerability Exploitation: Use Burp Suite to find and exploit SQL injection vulnerabilities in a sample website.
5. Use online OSINT tools (e.g., IPinfo.io, Shodan, Whois, DomainTools, Have I Been Pwned) to perform IP and domain information gathering, including geolocation, ownership, and registration details, and to identify data leaks or breaches linked to given email addresses.
6. ExifTool for Metadata Extraction: Using ExifTool, extract metadata from a given image and locate the geographic coordinates.
7. File Recovery with PhotoRec: Use PhotoRec to recover deleted files from a sample disk image.
8. USB History Investigation with Autopsy/FTK: Using Autopsy or FTK Imager, investigate USB history and identify any files that were transferred from or to the USB drive.
9. Set up two systems where one system generates continuous ICMP requests to another and observe the impact on system performance and availability over time.

Textbooks

1. P. Azad and P. Pathan, "Practical Cryptography: Algorithms and Implementations Using C++", CRC Press, 2021.
2. M. Sikorski and A. Honig, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software", No Starch Press, 2012.

Reference books

1. A. Jakobsen, "Practical Cyber Intelligence: A Hands-on Guide to Digital Forensics", 1st ed., Wiley, 2024.
2. D. Hook and J. Eaves, "Java Cryptography: Tools and Techniques", CRC Press, 2022.
3. S. J. Nielson and C. K. Monson, "Practical Cryptography in Python: Learning Correct Cryptography by Example", Apress, 2019.
4. S. Bowne, "Hands-On Cryptography with Python", Packt Publishing, 2018.
5. M. Welschenbach, "Cryptography in C and C++", 2nd ed., Apress, 2017.
6. J. Knudsen, "Java Cryptography", O'Reilly Media, 2011.
7. D. Stuttard and M. Pinto, "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws", 2nd ed., Wiley, 2011.

23PCIT603L Data Science Laboratory

Teaching Scheme:
Practical: 2 hours/week

Examination Scheme:
In-Semester:25 Marks
Oral: 25 marks
Credits: 1

Prerequisites: Basic programming, statistics, and linear algebra

Course Objectives:

Familiarize students with

1. Techniques for data wrangling and exploration
2. Modern tools for data visualization
3. Analysis of data for prediction and forecasting
4. Data science applications

Course Outcomes:

Students will be able to:

1. Implement techniques of data wrangling and exploration
2. Use modern tools for data visualization
3. Perform statistical analysis for prediction and forecasting
4. Develop a data science solution that provides meaningful data insights and predictions for real-world applications in the team

Suggested list of Assignments:

Capstone Project: End-to-End Data Science Application Development

Students will form groups and select a real-world problem domain to develop a data science solution that provides meaningful data insights and predictions. The project should use publicly available data or self-scraped data and demonstrate a working, reproducible application.

The project should incorporate the following core functionalities (at least 4):

1: Data Collection and Web Scraping

Collect data from a website or API using Python, BeautifulSoup, or Requests. Store in structured format and clean basic errors.

2: Data Cleaning and Preprocessing

Handle missing values, detect and treat outliers, and standardize data formats.

3: Exploratory Data Analysis (EDA)

Perform EDA using Python, Matplotlib, and Seaborn to identify patterns and trends.

4: Interactive Data Visualization and Dashboarding

Create a Power BI or Tableau dashboard with 4-5 interactive visuals.

5: Descriptive and Inferential Statistics

Compute descriptive statistics, correlation, and conduct hypothesis testing.

6: Time Series Forecasting

Apply ARIMA or Moving Average forecasting with model evaluation (MAE, RMSE).

Following advanced functionalities may be incorporated:

- Big Data Analysis
- Regression Analysis
- Classification Analysis

Expected Deliverables:

- Project Proposal (problem statement, objectives, data source)
- Cleaned Dataset & Scripts
- EDA Report and Visualizations
- Interactive Dashboard (Power BI/Tableau)
- Statistical & Predictive Model Reports
- Forecasting Results with Performance Metrics
- Final Capstone Report (PDF) + Demo Presentation

Textbooks

1. Wes McKinney, "Python for Data Analysis", 3rd Edition, O'Reilly Publication, ISBN: 9781098104030, 2022
2. Suresh Kumar Mukhiya, Usman Ahmed, "Hands-On Exploratory Data Analysis with Python" 1st Edition, 2020, Packt Publish, ISBN 978-1-80323-110-5

Reference Books

1. Ofer Mendelevitch, Casey Stella, Doug Eadline, "Practical Data Science with Hadoop and Spark: Designing and Building Effective Analytics at Scale, 1st Edition, Addison-Wesley, ISBN: 978-0134024141, 2017
2. Jeffrey Saltz and Jeffrey Stanton, "An Introduction to Data Science", 1st Edition, Sage Publications, ISBN: 978-1506377537, September 2017
3. J Jake Vanderplas, "Python Data Science Handbook", O'Reilly Publications, ISBN: 9781491912126, 2016
4. Montgomery, Douglas C.; Runger, George C., "Applied Statistics and Probability for Engineers" 2nd Edition, Wiley, ISBN 10: 0471170275, 1998
5. Gilbert Strang, "Introduction to linear algebra", 6th Edition, Wellesley-Cambridge Press, ISBN: 978-17331466-7-8, 2023

23PEIT601LA Deep Learning Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: 2 hours/week	**In-semester Exam:** 25 marks
	Oral: 25 marks
	Credits: 1

Prerequisites: Machine Learning

Course Objectives:

Familiarize students with:

1. Programming of deep learning
2. Libraries of deep learning
3. Types of deep learning techniques
4. Applications of deep learning

Course Outcomes:

Students will be able to:

1. Tune various hyperparameters to implement deep learning algorithms.
2. Apply CNN to image datasets.
3. Apply RNN and LSTM to sequential datasets.
4. Apply transformer architectures to textual datasets.

Suggested list of assignments:

Demonstrate use of drop out layers, optimizers, batch normalization etc.

1. Build a CNN to classify images from a dataset like CIFAR-10 or MNIST or a custom set of images (e.g., cats vs. dogs).
2. Implement a face detection system using libraries like OpenCV.
3. Use RNN and LSTM networks to forecast stock prices based on historical data.
4. Create a model that generates a textual description for an image, for example, using the Flickr8K dataset.

Additional list of assignments:

1. Develop a model that recognizes human activities from sensor data or video.
2. Use transformers / encoders - decoders to detect signs of Indian Sign Language.
3. Use an attention module to predict appropriate words in language models.
4. Design a gaming environment, integrate gamification elements, and train a Reinforcement Learning agent to successfully play the game (for example, tic-tac-toe, 4x4 sudoku etc.).

Textbooks:

1. Charu C. Aggarwal, "Neural Networks and Deep Learning", Springer 2nd Edition - 2023.
2. Ethem Alpaydin, "Introduction to Machine Learning", PHI 4th Edition – 2020.

Reference Books:

1. Nikhil Buduma, "Fundamentals of Deep Learning – Designing Next Generation Machine Intelligence Algorithms", O'Reily – 1st Edition – 2017.
2. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly 1st Edition 2021.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press 2016.
4. Francois Chollet, "Deep Learning with Python", Manning 2nd edition 2021.

Other Resources

1. UCI Machine Learning Repository https://archive.ics.uci.edu
2. WEKA Collection of datasets https://waikato.github.io/weka-wiki/datasets/
3. Kaggle datasets https://www.kaggle.com/datasets

23PEIT601LB Animation and Gamification Laboratory

Teaching Scheme:
Practical: 2 hours/week

Examination Scheme:
In-Semester: 25 Marks
Oral: 25 Marks
Credits: 1

Course Objectives:
Familiarize students with

1. Various methods and techniques used in Animation and Gamification
2. Applications of computer graphics in animation and gaming
3. Functions and libraries of OpenGL
4. Applications and implementation of computer graphics.

Course Outcomes:
Students should be able to

1. Create graphical representations using appropriate functions and algorithms.
2. Develop creative animations and interactive graphics for various applications.
3. Design gamified systems incorporating animated elements.
4. Design interactive stories and quiz games with integrated animations

Suggested list of assignments:

1. Study OpenGL environment, display-window management using GLUT, and OpenGL functions.
2. Write a program in OpenGL to draw a flag using any line drawing algorithm and fill it using scanline polygon filling algorithm.
3. Create a digital greeting card with simple animations (e.g., bouncing text, moving shapes).
4. Create a simple animation of weather elements like falling rain, moving clouds, or snow.
5. Build a game where a player moves a basket to catch falling objects with score.
6. Create a quiz game with animated responses for correct or incorrect answers, with leaderboard.

Open-ended assignment:
Develop a short interactive animation that changes scenes or character actions based on user inputs.

Textbooks

1. D. Hearn, M. Baker, "Computer Graphics with OpenGL", 4th Edition, Pearson Education, 2014, ISBN 978-93-325-1871-1
2. Andrew Hogue, Rick Parent, "Computer Animation Algorithms and Techniques", 3rd Edition , Morgan Kaufmann Publishers, 2012, ISBN 978-0124159730

Reference Books

1. G. Zichermann, C. Cunningham, "Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps", 1st Edition, O'Reilly Media, 2011, ISBN 978-1449397678.
2. J. Mathew, S. Nair, "Level Up Your Learning: The Power of Gamification", 1st Edition, Nova Science Publishers, 2024, ISBN 979-8891134096.
3. M. Dunsky, "C++ Game Animation Programming: Learn modern animation techniques with OpenGL and Vulkan" (2nd Edition), Packt Publishing, 2023, ISBN 978-1803246529.
4. M. Dunsky, "Mastering C++ Game Animation Programming", 1st Edition, Packt Publishing, 2025, ISBN 978-1835881927

23PEIT601LC Blockchain Laboratory

Teaching Scheme: | **Examination Scheme:**

Practical: 2 hours/week | **In-semester Exam:** 25 marks

Oral: 25 marks

Credits: 1

Prerequisites: Computer Networks

Course Objectives:

Familiarize students with:

1. Setup and operation of crypto wallets and blockchain transactions.
2. Smart contracts using Solidity in local and testnet environments.
3. Blockchain development tools, networks, and programming frameworks.
4. Decentralized applications integrating smart contracts and user interfaces.

Course Outcomes:

Students will be able to:

1. Configure different types of crypto wallets and perform blockchain transactions.
2. Implement smart contracts on local blockchain networks and testnets.
3. Examine blockchain tools, Bitcoin/Ethereum environments, and Solidity components.
4. Build a functional DApp using smart contracts and a frontend interface.

Suggested list of assignments

1. Crypto Wallet Configuration and Transactions
2. Local Blockchain Setup and Smart Contract Deployment
3. Solidity Programming: Arrays, Mappings, and Functions
4. Testnet Deployment and Documentation
5. DApp Development: Blockchain-based Voting System

Additional list of assignments

1. Bitcoin Core Exploration
2. ERC-20 Token Creation
3. NFT Minting Using ERC-721

Textbooks

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder Bitcoin and Cryptocurrency Technologies, Princeton University Press, 1st Edition, 2016
2. Chris Dannen, Introducing Ethereum and Solidity, Apress, 1st Edition, 2017

Reference Books

1. Andreas M. Antonopoulos & Gavin Wood, Mastering Ethereum, O'Reilly Media, 1st Edition, 2018
2. Andreas M. Antonopoulos, Mastering Bitcoin, O'Reilly Media, 2nd Edition, 2017
3. Imran Bashir, Mastering Blockchain, Packt Publishing, 4th Edition, 2023
4. Syed M. Haque, Solidity Programming Essentials, Packt Publishing, 2nd Edition, 2023

Other Resources

1. NPTEL: Blockchain and its Applications
2. NPTEL: Blockchain Architecture Design and Use Cases
3. https://buildspace.so/p/build-solidity-web3-app/lessons/welcome
4. https://www.theinsaneapp.com/2022/05/best-web3-projects.html
5. https://www.coinbase.com/learn/tips-and-tutorials/how-to-set-up-a-crypto-wallet

23VSEC601 Programming Skills in DevOps Laboratory

Teaching Scheme:	**Examination Scheme:**
Practical: 4 hours/week	**In-Semester:** 50 Marks
	Credits: 2

Prerequisites: Java programming skills
Course Objectives:
Familiarize students with

1. Insights of the DevOps environment
2. An overview of different DevOps tools
3. Continuous integration and continuous deployment
4. DevOps containerization and orchestration.

Course Outcomes:
Students will be able to:

1. Apply version control systems to manage and collaborate on projects effectively.
2. Automate unit and integration testing.
3. Build CI/CD pipelines for efficient application delivery.
4. Deploy applications using docker.

Mini Project:

Building an End-to-End DevOps Project
Students will form groups and will select a real-world problem to build a DevOps pipeline. Following points may be used as guidelines.

1. **Version Control with Git and GitHub**

   Set up Git and GitHub repositories and perform basic Git operations: clone, add, commit, push/pull, check in/check out, and create tags/branches. Collaborate using branches and merge changes. (Use Eclipse or VSCode IDE)

2. **JUnit**

   Test the application / prototype using unit and integration testing.

3. **Maven**

   Build a prototype of an application. Prepare unit test cases and execute.

4. **CI/CD Pipeline with Jenkins**

   Configure a Jenkins pipeline to build and prepare unit test cases and execute them. Integrate/API test stage with pipeline. Set up a post-build action to notify via email.

MKSSS'S CUMMINS COLLEGE OF ENGG. FOR WOMEN
KARVENAGAR
PUNE-52.

5. **Containerization with Docker**

   Write a Docker file to containerize a sample application. Build the docker image and run the application in a container. Push the docker image to docker Hub. Submit the Docker file, application running in a container screenshot, and docker Hub link.

Additional list of assignments:

1. **Deploy applications using Kubernetes.**

   Install and configure Minikube (or access a Kubernetes cluster). Write YAML files for Kubernetes deployments and services. Deploy a sample application to the Kubernetes cluster and expose it as a service. Submit the Kubernetes YAML files and screenshots of the deployed application.

2. **Configuration Management with Ansible**

   Write an Ansible playbook to configure a web server (e.g., Nginx or Apache). Test the playbook by applying it to a local or virtual server. Submit the Ansible playbook file and screenshots showing configuration results.

3. **Infrastructure as Code with Terraform**

   Write Terraform scripts to create resources on a cloud provider (e.g., AWS EC2 instance). Use variables and output values in the Terraform script. Apply and destroy the configuration. Provide Terraform scripts and screenshots of resources created on the cloud.

4. **SonarQube and Gating**

   Use SonarQube and Gating to scan the source code without executing the code to verify the code quality.

Textbooks

1. Ekamber K Singirikonda, DevOps Automation Cookbook, BPB Publications, 1st Edition, 2024.
2. Mikael Krief, Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins, 2nd Edition, March 2022

Reference Books

1. Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, Lynn Heidmann, "Introducing MLOps", O'Reilly Nov 2020
2. Billy Yuen, Jesse Suen, Alex Matyushentsev, Todd Ekenstam, "GitOps and Kubernetes", O'Reilly, April 2021
3. Gene Kim, Jez Humble, Patrick Debois, John Willis, and Nicole Forsgren, The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, 2nd Edition, October 2016

4. Len Bass, Ingo Weber, Liming Zhu, DevOps A Software Architect's Perspective, Pearson, May 2015.
5. Jennifer Davis and Katherine Daniels, "Effective DevOps", O'Reilly, 1stt Edition
6. Jez Humble, David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation"
7. Deepak Gaikwad, Viral Thakkar, "DevOps Tools from practitioner's Viewpoint", Wiley, 1st Edition
8. Aditya Pratap Bhuyan, "Docker Deep Dive: Learn, Build, and Scale"
9. Kelsey Hightower, Brendan Burns, and Joe Beda, "Kubernetes Up & Running: Dive into the Future of Infrastructure"
10. Sanjeev Sharma, The DevOps Adoption Playbook, A guide to adopting DevOps in a Multi-Speed IT Enterprise. Wiley, IBM Press, April 2017.

Other Resources:

1. https://azure.microsoft.com/en-in/solutions/devops/tutorial