Vishwakarma Institute of Technology Issue 01 : Rev No. 00 : Dt. 01/08/22

Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

NEP Compliant Structure & Syllabus of

Department of Computer Science & Engineering

(Internet of Things & Cyber Security Including Block Chain Technology)

Pattern 'A-24'

T. Y. B. Tech.

Effective from Academic Year 2026-27

Prepared by: - Board of Studies of CSE-IoTCSBT

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Chairman- BoS	Dean - Academics	Chairman – Academic Board
Dr. P M Shelke	Dr. P N. Mahalle	Dr. Rajesh Jalnekar

Contents

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

PSO1. Graduates will be able to design, implement, and optimize Internet of Things (IoT) systems.

PSO2. Graduates will be equipped to assess security vulnerabilities, design secure systems, and implement Blockchain-based distributed ledger solutions.

FF No: 653

Structure

Branch: CSE-IoTCSBT Year: Third Year (TY) Sem - I Academic Year 2026-27

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week))			Total	Credits
			Theory	Lab	Tut		
S1	CB3201	CYBER SECURITY	3	2	-	5	4
S2	CB3202	ADVANCED IOT AND EDGE COMPUTING	2	2	-	4	3
S3	CB3203A	ARTIFICIAL INTELLIGENCE	3	2	-	5	4
	CB3203B	CLOUD COMPUTING					
	CB3203C	HUMAN-COMPUTER INTERACTION					
	ID3001	BASICS OF GAME DEVELOPMENT					
	ID3002	MAINFRAME TECHNIQUE					
S4	MD3101	IBM FULL STACK DEVELOPER	-	-	-	-	3
	MD3102	META BACKEND DEVELOPER					
	MD3106	MICROSOFT POWER BI DATA ANALYST					
	MD3113	GOOGLE DATA ANALYTICS					
	MD3121	IBM DEVOPS AND SOFTWARE ENGINEERING					
	ID3004	DEVOPS AWS SKILL BUILDER-1					
S5	MD3201	FINTECH	-	-	-	-	4
	MD3202	DIGITAL MARKETING					
	MD3203	ENTREPRENEURSHIP					
	MD3204	MANAGEMENT					
S6	MM0501A	OPERATING SYSTEMS AND DISTRIBUTED SYSTEMS	2	-	1	3	3
	MM1501D	PRODUCT LIFECYCLE MANAGEMENT					
S7	CB3205	DESIGN THINKING-3	-	-	1	1	1
	IR3101	INTERNATIONAL SKILL DEVELOPER MODULE					
S8	CB3204	ENGINEERING RESEARCH AND INNOVATION-1	-	-	-	-	2
	IR3102	INTERNATIONAL ENGINEERING DEVELOPMENT PROGRAM					
Total			**10**	**6**	**2**	**18**	**24**

Chairman- BoS
Dr. P M Shelke

Dean - Academics
Dr. P N. Mahalle

Chairman – Academic Board
Dr. Rajesh Jalnekar

S5 Open Elective-1 (Coursera)			
FinTech-1	**Digital Marketing-1**	**Entrepreneurship-1**	**Management-1**
FinTech: Finance Industry Transformation and Regulation Specialization	Foundations of Digital Marketing and E-commerce	Essentials of Entrepreneurship: Thinking & Action	Digital Product Management Specialization
Digital Transformation in Financial Services Specialization	Adobe Digital Marketing	Technology & Entrepreneurship Specialization	Project Management: Tools, Approaches, Behavioural Skills Specialization
Financial Markets Specialization	Digital Marketing Operations & Analytics Specialization	Entrepreneurship: Growing Your Business Specialization	Foundations of Management Specialization
FinTech Law and Policy	Social Media and Digital Marketing Fundamentals	Skills for a Modern Entrepreneur or Business Owner Specialization	Supply Chain Management Specialization
Decentralized Finance (DeFi): The Future of Finance Specialization	Introduction to Digital Marketing	Innovation & Entrepreneurship - From Design Thinking to Funding	
	Digital Marketing Trends	Innovation and entrepreneurship Specialization	
	Assess for Success: Marketing Analytics and Measurement	The Entrepreneurial Toolkit Specialization	
	Concepts, Strategies, and Analytics in Performance Marketing and Digital Advertising Specialization		
	Integrated Marketing Communications: Advertising, Public Relations, Digital Marketing and more		
	Marketing in a Digital World		
	Marketing Analytics Foundation		

Chairman- BoS
Dr. P M Shelke

Dean - Academics
Dr. P N. Mahalle

Chairman – Academic Board
Dr. Rajesh Jalnekar

FF No. 653_A

Examination scheme

Branch: CSE-IoTCSBT

Academic Year: 2026-27 **Semester: I** **Module: 5**

Pattern: A24

Level 5.5																								
Course Code	NEP Course Category	Course Name	Teaching Scheme (Hrs./week)			Examination Scheme and Marks														Credits				
			Theory	Tutorial	Practical	CVV	CP	LAB/ TUT	GD/ PPT	HA	MSE (W)	MSE (R)	MSE (O)	ESE(R)	ESE (W)	ESE (O)	PRACT+ CVV	ESE-Direct Evaluation		Theory	Tutorial	Practical	Total	
Semester I																								
CB3201	PCC	Cyber Security	3	-	2										40		60			3	-	1	4	
CB3202	PCC	Advanced IoT and Edge Computing	2	-	2	20	30	10							40					2	-	1	3	
CB3203A	PEC	Artificial Intelligence	3	-	2	20	40		20	20										3	-	1	4	
CB3203B		Cloud Computing																						
CB3203C		Human Computer Interaction																						
ID3001		Basics of Game Development																						
ID3002		Mainframe Technique																						
MD3101	PEC (COURSERA)	IBM Full Stack Developer	-	-	-													100		-	-	-	3	
MD3102		Meta Backend Developer																						
MD3106		Microsoft Power BI Data Analyst																						
MD3113		Google Data Analytics																						
MD3121		IBM Devops And Software Engineering																						
ID3004		DevOps AWS Skill Builder-1																						
MD3201	OE (COURSERA)	Fintech	-	-	-													100		-	-	-	4	
MD3202		Digital Marketing																						
MD3203		Entrepreneurship																						
MD3204		Management																						
MM0501A	MDM	Operating Systems and Distributed Systems	2	1	-		30		20	10					40					2	1	-	3	
MM1501D		Product Lifecycle Management				20	30		10								40						4	3+1DT
CB3205	VSEC	Design Thinking-3	-	1	-													100		-	1	-	1	
IR3101		International Skill Developer Module																						
CB3204	FP	Engineering Research and Innovation-1	-	-	-															-	-	-	2	
IR3102		International Engineering Development Program										30		70										
Total			10	2	6	60	130	10	50	30	0	30	0	70	120	0	100	300		10	2	3	24	

Abbreviations Used	
PPT	Powerpoint Presentation
GD	Group Discussion
CVV	Comprehensive Viva Voce
LAB	Laboratory
TUT	Tutorial
CP	Course Project
PR	Practical
HA	Home Assignment
MSE	Mid Semester Examination
ESE	End Semester Examination
R /W/O	Review /Written/Online

FF No. : 654

CB3201: Cyber Security

Credits: 4 **Teaching Scheme**: Theory: 3 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

1. Data Communication & Networking
2. Computer Network

Course Objectives:

1. To understand the basics of cyber security and its importance
2. To identify and analyze security threats and vulnerabilities
3. To learn about security controls and risk management
4. To apply security best practices in real-world scenarios
5. To impart fundamental knowledge of securing network infrastructures and cloud computing environments
6. To familiarize students with the legal landscape of cyberspace

Relevance of this course:

The Cyber Security course provides students with knowledge of cyber threats, security techniques, and risk management to protect digital systems and data. It develops practical skills in network security, cryptography, ethical hacking, and digital forensics. The course also creates awareness of legal, ethical, and professional responsibilities, preparing students for careers in cyber security and information security.

SECTION I

Unit 1: Introduction to Cyber Security (7 Hours)

Overview of cyber security, Importance of cyber security, Types of cyber threats (malware, phishing, ransomware), Security threats and vulnerabilities, Cyber security frameworks and standards.

Unit 2: Security Threats and Vulnerabilities (7 Hours)

Types of malwares (viruses, worms, trojans), Phishing and social engineering attacks, Vulnerability assessment and penetration testing, Threat modelling and risk assessment

Unit 3: Security Controls and Risk Management (7 Hours)

Access control (authentication, authorization, accounting), Encryption (symmetric and asymmetric), Firewalls and intrusion detection systems, Risk management and mitigation strategies

SECTION II

Unit 4: Web Application Security and Cryptography (7 Hours)

Web application vulnerabilities (SQL injection, cross-site scripting), Secure coding practices, Web application firewalls, Cryptography basics (hash functions, digital signatures)

Unit 5: Network Security Architecture (7 Hours)

Understanding firewalls (packet filtering, stateful inspection), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) and the concept of "Defense in Depth", Cloud Computing Security, Wireless and Mobile Security

Unit 6: Cyber Law, Forensics, and Ethics (7 Hours)

Introduction to Digital Forensics, Cyber Laws and IT Act, Incident Response and Recovery, Cyber Ethics and Professional Responsibility

List of Practical:

1. Cyber security Dos & Don'ts, cyber security awareness, government CS portals
2. Introduction to Nmap: Scanning for open ports and services
3. Vulnerability Scanning using OpenVAS: Identifying vulnerabilities in a system
4. Password cracking using John the Ripper: Understanding password security
5. Firewall Configuration: Configuring a firewall to block incoming traffic
6. Intrusion Detection using Snort: Detecting and alerting on suspicious traffic
7. Encryption using OpenSSL: Encrypting and decrypting files using symmetric and asymmetric encryption
8. Web Application Security Testing using OWASP ZAP: Identifying vulnerabilities in a web application.
9. SQL Injection Attack: Understanding SQL injection vulnerabilities
10. Cross-Site Scripting (XSS) Attack: Understanding XSS vulnerabilities
11. Secure Coding Practices: Implementing secure coding practices in a programming project
12. Digital Signature using GPG: Creating and verifying digital signatures
13. Network Traffic Analysis using Wireshark: Analyzing network traffic to identify security issues.
14. Case studies of Cybercrime and online banking frauds.

List of Project Areas:

1. Intrusion Detection System (IDS) Implementation.
2. Automated Firewall Management
3. Secure Password Manager with Advanced Crypto.
4. Static Application Security Testing (SAST) Tool.

Text Books:

1. M. E. Whitman and H. J. Mattord, *Cybersecurity Fundamentals*. Boston, MA, USA: Cengage Learning, 2021.

2. C. P. Pfleeger, S. L. Pfleeger, and J. Margulies, *Security in Computing*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2015.

Reference Books:

1. Online security resources (e.g., OWASP, SANS Institute)
2. Industry reports and whitepapers on cyber security

e Learning Resources:

NPTEL

1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs144
   Cyber Security and Privacy, By Prof. Saji K Mathew
2. https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge86 - Cyber Security, Tools, Techniques and Counter Measures, By Prof. Dr. Nilesh K Modi

Course Outcomes:

The student will be able to –

CO1: Understand the concepts of cyber security and its importance
CO2: Identify and analyze security threats and vulnerabilities
CO3: Apply security controls and risk management techniques
CO4: Implement security best practices in real-world scenarios
CO5: Analyze various network security components like Firewalls, IDS, and IPS
CO6: Interpret legal provisions under the Indian IT Act and global data privacy laws to ensure organizational compliance.

CO-PO Mapping Matrix: Fundamentals of Data Structures

CO	Program Outcomes (PO)										PSO			
CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO 11	PO12	PSO 1	PSO2
CO1	3	2	-	-	1	3	1	2	-	-	-	2	-	2
CO2	2	3	-	2	2	2	-	2	-	-	-	1	-	3
CO3	2	3	3	2	3	2	-	2	-	-	-	1	1	3
CO4	2	2	3	2	3	3	1	3	2	2	-	2	2	3
CO5	3	3	2	2	3	2	-	1	-	-	-	1	2	3
CO6	1	2	-	-	-	3	2	3	-	2	-	2	-	2
Average	2.17	2.5	2.67	2.0	2.4	2.5	1.33	2.17	2.0	2.0	0	1.5	1.67	2.67

Scale:

3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping for all COs

CO1: Understand the concepts of cyber security and its importance

- **PO1 (3):** Students learn engineering fundamentals related to information security and cyber threats.
- **PO2 (2):** Basic understanding enables identification of security-related problems.
- **PO5 (1):** Introduces security tools and technologies used in cybersecurity.
- **PO6 (2):** Explains societal and legal importance of cybersecurity.
- **PO7 (1):** Highlights cybersecurity's role in sustainable digital infrastructure.
- **PO8 (2):** Introduces ethical responsibilities in handling information.
- **PO12 (2):** Encourages continuous learning due to evolving cyber threats.
- **PSO2 (2):** Establishes the foundation for secure system and blockchain security concepts.

CO2: Identify and analyze security threats and vulnerabilities

- **PO1(2):** Uses engineering fundamentals to understand attacks.
- **PO2 (3):** Focuses on analyzing vulnerabilities and security risks.
- **PO4 (2)**: Involves investigation of threats using analytical methods.
- **PO5 (2):** Uses vulnerability assessment and scanning tools.
- **PO6 (2):** Evaluates security risks affecting organizations and society.
- **PO8 (2):** Considers ethical issues in security testing.
- **PO12 (1):** Encourages updating knowledge of emerging threats.
- **PSO2 (3):** Directly supports vulnerability assessment and secure system design.

CO3: Apply security controls and risk management techniques

- **PO1 (2):** Applies engineering concepts to security implementation.
- **PO2 (3):** Analyses risks before selecting controls.
- **PO3 (3):** Designs appropriate security solutions and mitigation strategies.
- **PO4 (2):** Evaluates effectiveness of implemented controls.
- **PO5 (3):** Uses modern security tools for implementing controls.
- **PO6 (2**): Applies organizational security policies and compliance requirements.
- **PO8 (2):** Ensures ethical handling of organizational assets.
- **PO12 (1):** Promotes continuous learning in risk management.
- **PSO1 (1):** Security controls are applicable in IoT environments.
- **PSO2 (3):** Directly supports secure system development and blockchain security.

CO4: Implement security best practices in real-world scenarios

- **PO1 (2):** Applies cybersecurity fundamentals in practical environments.
- **PO2 (2):** Analyses practical security requirements.
- **PO3 (3**): Implements secure solutions and configurations.
- **PO4 (2**): Tests and validates implemented security practices.
- **PO5 (3):** Uses firewalls, antivirus, SIEM, IDS/IPS, and other security tools.
- **PO6 (3):** Addresses legal and organizational responsibilities.
- **PO7 (1):** Promotes sustainable and secure computing practices.
- **PO8 (3):** Emphasizes professional ethics and responsible security practices.
- **PO9 (2):** Security implementation often requires teamwork.
- **PO10 (2):** Students prepare reports and presentations on implementation.
- **PO12 (2):** Encourages continuous learning of emerging best practices.
- **PSO1 (2):** Best practices improve IoT system security.
- **PSO2 (3):** Strongly contributes to secure system implementation.

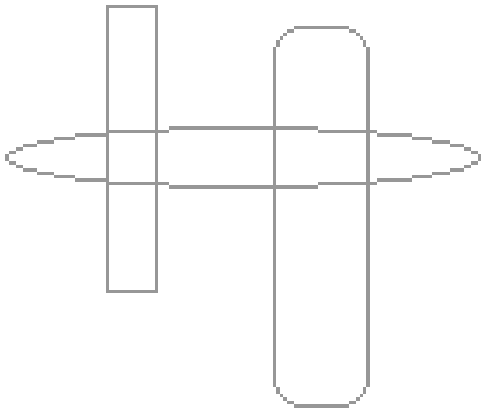
CO5: Analyze various network security components like Firewalls, IDS, and IPS

- **PO1 (3):** Requires networking and engineering knowledge.
- **PO2 (3):** Analyses network attacks and defense mechanisms.
- **PO3 (2):** Selects appropriate security architectures.
- **PO4 (2):** Investigates effectiveness of network security devices.
- **PO5 (3**): Uses modern network security tools and appliances.
- **PO6 (2):** Ensures secure networking for organizational environments.
- **PO8 (1):** Promotes ethical deployment of security technologies.
- **PO12 (1):** Encourages learning emerging network defense technologies.
- **PSO1 (2):** Network security is essential for IoT communication.
- **PSO2 (3)**: Directly supports secure network and blockchain infrastructure.

CO6: Interpret legal provisions under the Indian IT Act and global data privacy laws to ensure organizational compliance.

- **PO1 (1):** Requires basic understanding of cybersecurity principles.
- **PO2 (2):** Analyses legal implications of cyber incidents.
- **PO6 (3):** Directly addresses legal, societal, and professional responsibilities.
- **PO7 (2):** Promotes sustainable governance and responsible use of technology.

- **PO8 (3):** Emphasizes ethics, privacy, and professional conduct.
- **PO10 (2):** Requires communicating legal compliance and security policies.
- **PO12 (2):** Encourages continuous learning of evolving cyber laws.
- **PSO2 (2):** Supports secure and legally compliant system development.



FF No. : 654

CB3202: Advanced IOT and Edge Computing

Credits: 3 **Teaching Scheme:** Theory: 2 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

1. Data Communication and Networking
2. Basics of IOT

Course Objectives:

1. To understand advanced IOT and technologies in its conjunction.
2. To study the concept of virtualization and its application to sensors.
3. To study various hardware related to IOT.
4. To understand applicability of AI in Edge and Edge in AI

Course Relevance:

This course provides knowledge of designing and deploying smart IoT systems integrated with edge computing for real-time data processing and decision-making. It equips students with skills in sensor networks, cloud-edge integration, data analytics, security, and AI-enabled IoT applications. The course is highly relevant to Industry 4.0, smart cities, healthcare, agriculture, and industrial automation, preparing students for careers in IoT, embedded systems, cloud computing, and edge AI.

SECTION I: ADVANCED IOT, VIRTUALIZATION AND SENSORS

Unit 1: Advanced IOT and Cloud Computing (7 Hours)

IOT Analytics, Types, Platforms, ThingsSpeak, Google Analytics for Firebase, Statistical analysis using excel and BI tools, Amazon IOT technologies, Microsoft IOT technologies, IIOT applications, cloud computing basics, cloud computing architecture, cloud computing technologies Map reduce, Hadoop, file system, AWS platform, Google analytics platform.
Case study: Home automation system using Firebase and ThingsSpeak platform.

Unit 2: Virtualization and Sensors (7 Hours)

Virtualization, basic concepts, types, Hardware Virtualization, Operating system Virtualization, Server Virtualization, Storage Virtualization, Level of Virtualization, Virtualization Structure/Tools and Mechanisms, benefits, advantages, layers of virtual sensors, complex event processing, virtual sensing, enabling technologies, Sensing and sensor-related technologies, Cyber security, IoT, Digital twins/triplets, Quantum computing, Physical Versus Virtual Sensors, Virtual Sensor characteristics, Applications of virtual sensing, demand resource optimization.

SECTION II: IOT HARDWARE, EDGE AND DRONE TECHNOLOGY

Unit 3: IOT Hardware and Simulation (7 Hours)

Hardware Specification, adder, subtractor, CMOS, MOSFET, PCB Manufacturing Process, Automotive sensors (Ultrasonic sensor, Radar, Lidar, Camera), 3D Magnetic Sensors, Wireless Robotic Arm 2.4, Solid State Relay and its types, ULN2003 Stepper Motor Driver – Circuit,Nvidia

Jetson Nano, V-rep simulator, ADAS, Vehicle Architectures, FGPA, Hardware design for IoT projects, Icestudio, FireSim**,** HDL simulators, ISim, ModelSim. Protocols Simulators.
Case Study: Simulation of virtual testbeds using coopelia simulator.

Unit 4: Edge Computing, Artificial Intelligence and Drone Technology (7 Hours)

Edge Computing: Introduction, basics, evolution, Vertical associations on industrial automation. Functional mapping of edge computing activities. Fundamentals of Artificial Intelligence and Machine Learning: Artificial Intelligence, Machine Learning, Explainable AI for Machine Learning and Edge Computing
Case study: Virtualization of motor temperature sensor.
Drone Technology: Introduction, History of UAV, Components Selection, AirFrame Selection and Integration, Transmitter Receiver, Control and Testing, Applications of Drone system. Drone variants.

List of Practical:

1. Conduct visualization and analytics on Power BI platform
2. Installation and configuration of Hadoop Cloud.
3. Hello world using Map-reduce.
4. Install and configure Google App Engine.
5. Study the Mbed API for simulated microcontrollers.
6. Wireless Robotic Arm 2.4 based pickup and drop system.
7. Fast Object Detector using Nvidia Jetson Nano
8. Embedded ML for Hard Hat detection using edge Impulse studio
9. Anomaly Detection using edge impulse studio
10. A study assignment on Drones: Unmanned Aerial Vehicles and Their Impact

List of Project areas:

1. Autonomous Vehicles
2. Virtual Sensors
3. Drone Technology
4. Edge Computing

Text Books:

1. Vermesan, Ovidiu, and Peter Friess, eds. Internet of things: converging technologies for smart environments and integrated ecosystems. River publishers, 2013.
2. Wang, Xiaofei, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. "Edge AI: Convergence of edge computing and artificial intelligence." Singapore (2020).
3. Kapoor, Amita. Hands-On Artificial Intelligence for IoT: Expert machine learning and deep learning techniques for developing smarter IoT systems. Packt Publishing Ltd, 2025.

Reference Books:

1. Velte, Anthony T., Toby J. Velte, Robert C. Elsenpeter, and Robert C. Elsenpeter. "Cloud computing: a practical approach." (2010).
2. Khang, Alex, Vugar Abdullayev, Vladimir Hahanov, and Vrushank Shah, eds. Advanced IoT technologies and applications in the industry 4.0 digital economy. CRC Press, 2024.
3. Manoharan, Kalaiselvi Geetha, Jawaharlal Arun Nehru, and Sivaraman Balasubramanian. "Artificial intelligence and IoT." Springer 10 (2021): 978-981.

E-Learning Resources:

NPTEL

1."Advanced IOT Applications", https://www.nptelprep.in/courses/108108123 . The course introduces advanced applications of IoT , computer vision, and vehicle patrolling.

Coursera

2."Advanced IoT Systems Integration and Industrial Applications", https://www.coursera.org/learn/advanced-iot-systems-and-industrial-applications-course-3 . The course provides introduction to advanced topics related to IoT. These include edge computing, IoT security and combining IoT into various industrial applications.

Swayam

3. "AI driven Perception, Learning and Mapping for Drones" , https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ee96 . The course covers basics of ANN, ML and its specific applications for drones. Search for "AI driven Perception, Learning and Mapping for Drones" at https://swayam.gov.in/.

Course Outcomes:

The student will be able to –

CO1: Describe the practical application of IOT based analytics.

CO2: Apply the concept of sensors and virtualization.

CO3: Understand different types of IOT hardware devices and their interfacing.

CO4: Explain the integration of artificial intelligence and edge computing.

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)												PSO	
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO 11	PO 12	PSO 1	PSO 2
CO1	3	2			3						2			
CO2												2	2	
CO3	2		2		2							1		
CO4						3			2		2		1	
Average	2.00	1.50	0.75	1.50	0.00	0.75	0.00	0.00	0.75	0.00	0.20	0.30	0.30	0.00

Scale:

3 – Strongly Mapped

2 – Moderately Mapped

1 – Slightly Mapped

Blank – Not Mapped

Justification for Mapping

CO1: Describe the practical application of IOT based analytics.

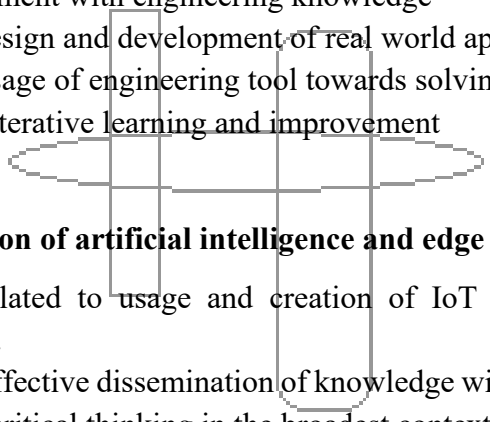
- **PO1(3)**: strong alignment with engineering knowledge
- **PO2(2):** related to analysis of systems
- **PO5(3)**: strong alignment to modern tools and their usage.
- **PO11(2):** related to continuous upgradation of knowledge and its learnings

CO2: Apply the concept of sensors and virtualization.

- **PO12(2)**: related to iterative learning and improvement
- **PSO1(2):** strongly related to creation and design of IoT systems

CO3: Understand different types of IOT hardware devices and their interfacing.

- **PO1(2)**: strong alignment with engineering knowledge
- **PO3(2)**: related to design and development of real world applications
- **PO5(2):** related to usage of engineering tool towards solving problems in real world
- **PO12(1):** related to iterative learning and improvement



CO4: Explain the integration of artificial intelligence and edge computing.

- **PO6(3):** strongly related to usage and creation of IoT based devices for societal and environmental needs.
- **PO9(2):** related to effective dissemination of knowledge within the engineering community.
- **PO11(2):** related to critical thinking in the broadest context of technological change.
- **PSO1(1):** related to creation of IoT technologies.

FF No. : 654

CB3203A: Artificial Intelligence

Credits: 4 **Teaching Scheme:** Theory: 3 Hours/Week, Lab: 2Hours/week

Prerequisites:

1. Fundamentals of Programming
2. Data Structures and Algorithms

Course Objectives:

1. To understand the key characteristics and types of intelligent agents.
2. To explore uninformed and informed search strategies.
3. To develop knowledge representation techniques and reasoning under uncertainty.
4. To understand planning, expert systems, and machine learning integration in AI.
5. To explore applications of AI in NLP, CV, robotics, and societal domains.
6. To introduce advancements including Explainable AI (XAI), GenAI, and cloud platforms.

Course Relevance:

This course provides a strong foundation in Artificial Intelligence, including intelligent agents, search algorithms, machine learning, and knowledge representation. It enables students to develop AI solutions for applications such as NLP, computer vision, robotics, and expert systems. The course also emphasizes Responsible AI, Explainable AI (XAI), and ethical practices, preparing students for careers in AI and intelligent systems.

SECTION I: Foundations Of Intelligence and Knowledge Representation

Unit 1: Introduction to AI and Intelligent Agents (7 Hours)

Definition/evolution of AI; Human vs. Machine Intelligence; Agent types and environments; Problem formulation; State-space representation; Uninformed vs. Informed Search; Ethical, Responsible, and Explainable AI (XAI); GenAI overview; Case Studies: AI for Smart Cities.

Unit 2: Search Strategies and Game Playing (7 Hours)

Uninformed search (BFS, DFS, UCS); Informed search (A*, Greedy, Beam Search, Hill Climbing); Game Playing (Minimax, Alpha-Beta Pruning); Optimization via search: evolutionary algorithms and swarm intelligence; Examples: Maze-solving, game bots.

Unit 3: Knowledge Representation and Inference (7 Hours)

Propositional/Predicate/First Order Logic; Resolution; Semantic networks; Frames; Ontologies, RDF, OWL; Knowledge Graphs; Forward/Backward Chaining; Tools: Neo4j, SPARQL basics.

SECTION II: Advanced AI Applications

Unit 4: Planning & Learning in Expert Systems (7 Hours)

Classical/Hierarchical Planning; Goal Stack Planning; STRIPS; Uncertain Planning: Markov Decision Processes (MDP); Transfer Learning (BERT, ResNet); Integration with ML pipelines; Applications: Traffic control, cybersecurity, MLOps.

Unit 5: AI Applications (7 Hours)

NLP: NER, Summarization, Transformers (BERT, GPT); Computer Vision: Defect Detection, Surveillance, Gesture Recognition; Speech Recognition/Assistive Tech; Robotics: Perception, Planning, Navigation; Multimodal AI (Gemini, GPT-4V).

Unit 6: Advancements of AI on Cloud Platforms (7 Hours)

Cloud platforms: AWS (SageMaker), Google Vertex AI, Azure ML; Low-code tools: Teachable Machine; MLOps: GitHub Actions, Docker, Monitoring; Project deployment using cloud APIs.

List of Practical:

1. Classify real-world agents (e.g., Alexa, Tesla) into reflex, goal-based, utility-based, or learning agents.
2. Implement BFS and A* for solving an 8-puzzle or a maze. Compare performance.
3. Design heuristic functions for a maze and implement Greedy Best-First and Hill Climbing algorithms.
4. Develop a Tic-Tac-Toe game AI using Minimax algorithm with Alpha-Beta pruning.
5.Solve a Constraint Satisfaction Problem (CSP) like Sudoku or Graph Coloring using Backtracking with Forward Checking.
6. Create a semantic network and a frame representation for a simple scenario like a family tree. Demonstrate forward and backward chaining.
7. Plan a simple AI agent using Goal Stack Planning (e.g., Block World).
8. Perform Named Entity Recognition using Hugging Face Transformers on sample text data.
9.Construct a Knowledge Graph (using Neo4j or SPARQL) for a real-world domain like university data.
10. Mini-project: Build a simple AI-based chatbot or image classifier using cloud platforms and deploy it using MLOps best practices.

List of Project Areas:

1. Intelligent Search & Game AI: Autonomous maze-solving agents using A* and adversarial game bots (e.g., Chess, Connect4) using Alpha-Beta Pruning.
2. Knowledge-Based Systems: Domain-specific Knowledge Graphs (e.g., University or Healthcare data) using Neo4j or SPARQL.
3. Natural Language Processing (NLP): AI-based Chatbots or text summarization tools using Hugging Face Transformers (BERT/GPT).
4. Computer Vision & Perception: Industrial Defect Detection, real-time Gesture Recognition, or smart surveillance systems.

5. Autonomous Systems & IoT: Adaptive Traffic Control using AI planning (STRIPS/MDP) and obstacle avoidance for mobile robots.

Course Outcomes:

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

CO1: Explain key AI concepts, intelligent agent architectures, and ethical considerations including Responsible AI and XAI.

CO2: Implement various AI search algorithms, including uninformed, informed, and adversarial strategies.

CO3: Design and apply knowledge representation schemas and logical inference techniques for reasoning.

CO4: Demonstrate planning techniques and integrate machine learning approaches in expert systems.

CO5: Apply AI techniques for real-world applications in domains like NLP, computer vision, and robotics.

CO6: Understand best practices for scalability, reliability, and security in cloud-based AI solutions.

Text Books:

1. S. Russell and P. Norvig; *Artificial Intelligence: A Modern Approach*; 3rd ed., Prentice Hall.
2. E. Rich, K. Knight, and S. B. Nair; *Artificial Intelligence*; 3rd ed., Tata McGraw Hill (TMH).
3. P. H. Winston; *Artificial Intelligence*; 3rd ed., Addison-Wesley Publishing Company

Reference Books:

1. S. Kausik; *Artificial Intelligence*; 1st ed., Cengage Learning.
2. N. P. Padhy; *Artificial Intelligence and Intelligent Systems*; 1st ed., Oxford University Press.
3. D. Khemani; *A First Course in Artificial Intelligence*; 1st ed., McGraw Hill Education (India).
4. I. Bratko; *Prolog Programming for A.I.*; 4th ed., Pearson (distributed via TMH).

e Learning Resources:

NPTEL-

1. "Artificial Intelligence (Autumn 2020)", Prof. Pallab Dasgupta, IIT Kharagpur, https://cse.iitkgp.ac.in/~pallab/artificial_intelligence_autumn_2020/index.html
2. "Artificial Intelligence: Search Methods for Problem Solving", Prof. Deepak Khemani, NPTEL (IIT Madras), https://onlinecourses.nptel.ac.in/noc22_cs56/preview
3. "Introduction to Artificial Intelligence", Prof. Mausam, NPTEL (IIT Delhi), https://onlinecourses.nptel.ac.in/noc23_ge40/preview

edX –

1. CS50's Introduction to Artificial Intelligence with Python (Harvard University): Covers graph search, reinforcement learning, and neural networks using Python.

2. AI for Everyone (IBM): A non-technical introduction to AI concepts and project workflows

YouTube –

1. DeepLearning.AI (Andrew Ng): The famous "Machine Learning Specialization" lectures and "Deep Learning Specialization" are available for free viewing.
2. MIT OpenCourseWare (OCW): Search for "MIT 6.034 Artificial Intelligence" to see Patrick Winston's original lectures

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)												PSO	
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO 12	PSO1	PSO2
CO1	2	1	-	-	-	3	3	-	-	-	-	2	2	-
CO2	3	3	2	2	2	-	-	-	-	-	-	-	3	1
CO3	2	3	3	1	2	-	-	1	-	-	-	-	2	2
CO4	2	2	3	2	3	-	-	2	-	1	-	1	2	3
CO5	1	2	3	3	3	2	-	3	3	-	-	2	3	3
CO6	1	-	2	-	3	1	-	2	2	-	-	3	1	3
Average	1.83	2.2	2.6	2	2.6	2	3	2	2.5	1	-	2	2.17	2.4

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

CO1: Explain key AI concepts, intelligent agent architectures, and ethical considerations including Responsible AI and XAI.

- **PO1(2):** Fundamental knowledge of AI evolution and agent types (Reflex, Goal, Utility).
- **PO6/PO7(3):** High relevance to societal impact and mandatory study of **Responsible AI** and **XAI**.
- **PO12(2):** Promotes independent learning by introducing emerging concepts like **GenAI**.

CO2: Implement various AI search algorithms, including uninformed, informed, and adversarial strategies.

- **PO1/PO2(3):** Application of mathematical principles and search space theory to formulate strategies.
- **PO3/PO4(2):** Design of heuristic functions and investigation of algorithm performance (time/space complexity).

- **PO5(2):** Use of programming tools to implement search-based agents.

CO3: Design and apply knowledge representation schemas and logical inference techniques for reasoning.

- **PO2/PO3(3):** Analytical evaluation and design of structured ontologies, frames, and inference systems.
- **PO5(2):** Use of modern tools like **Neo4j** and **SPARQL** for Knowledge Graphs.
- **PO8(1):** Collaborative work during construction of knowledge schemas.

CO4: Demonstrate planning techniques and integrate machine learning approaches in expert systems.

- **PO1/PO2(2):** Theoretical knowledge of **STRIPS** planning and analysis of **Markov Decision Processes (MDP)**.
- **PO3/PO5(3):** Design contribution via goal-stack planning and use of **ML pipelines**.
- **PO10(1):** Introductory management principles via MLOps and traffic control applications.

CO5: Apply AI techniques for real-world applications in domains like NLP, computer vision, and robotics.

- **PO3/PO4/PO5(3):** Designing solutions for tasks like **NER** and investigating pre-trained models like **Transformers (BERT/GPT)**.
- **PO8/PO9(3):** Multi-disciplinary team development and effective technical project

CO6: Understand best practices for scalability, reliability, and security in cloud-based AI solutions.

- **PO5(3):** High reliance on cloud platforms (**AWS SageMaker**, **Vertex AI**) and **Docker/MLOps**.
- **PO12(3):** Recognition of the need for lifelong learning in rapidly changing cloud-AI technologies.

FF No. : 654

CB3203B: Cloud Computing

Credits: 4 **Teaching Scheme:** Theory: 3 Hours/Week, Lab: 2 Hours/Week

Prerequisites:

1. Operating Systems
2. Computer Networks
3. Database Management System

Course Objectives:

1. To become familiar with cloud computing and its ecosystem
2. To acquire basics of virtualization and its importance
3. To Evaluate different cloud storage solutions (object, block, and file storage) to determine their suitability for various real-world applications.
4. To Examine and evaluate Continuous Integration and Continuous Delivery (CI/CD) practices for improving software quality and deployment efficiency.
5. To Examine and design secure identity and access control mechanisms using IAM policies, roles, MFA, and federated access.
6. To Explore the emerging trends in the Cloud computing environment.

Course Relevance:

Cloud Computing enables scalable, cost-effective, and secure IT solutions for modern organizations. This course equips students with skills in virtualization, cloud services, storage management, networking, CI/CD automation, and identity management, while introducing emerging cloud technologies. These competencies prepare learners for careers in cloud engineering, DevOps, cloud administration, and solution architecture.

SECTION I

Unit 1: Introduction to Cloud Computing (7 Hours)

Recent trends in computing, Cluster computing, Distributed computing, Evolution of cloud computing, Cloud versus traditional architecture, Cloud Computing Architecture, Google Cloud architecture, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Public cloud, Private cloud, Hybrid cloud, Community cloud

Unit 2: Virtualization (7 Hours)

Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM), Compute options in the cloud, Exploring IaaS with Compute Engine, Configuring elastic apps with auto scaling, Basics of virtualization and implementation challenges.

System virtualization technologies-architectures and internals. KVM, Xen, VMware. Amazon Elastic Compute Cloud EC2 as computing service. Memory virtualization-virtualization techniques, ballooning, deduplication and sharing. Network and storage virtualization, Virtual machine migration and replication techniques pre-copy and post-copy techniques, applicability to system availability.

Unit 3: Cloud Infrastructure, & Cloud Database (7 Hours)

Virtual machines, containers, and serverless computing, Storage solutions: Object storage, block storage, file storage, Cloud networking: VPC, Load Balancers, DNS, CDN, Hands-on: Setting up a cloud instance and networking, Virtual machines vs. containers vs. serverless.
Cloud databases: SQL (RDS, Cloud SQL), NoSQL (DynamoDB, CosmosDB), Data backup and disaster recovery strategies, Configuring and querying a cloud database.

SECTION II

Unit 4: DevOps (7 Hours)

Introduction to DevOps: DevOps culture and principles, Benefits of DevOps, Comparison with traditional software development methodologies. Continuous Integration and Continuous Delivery (CI/CD): CI/CD pipeline overview, Tools used in CI/CD (Jenkins, GitHub Actions, GitLab CI/CD, CircleCI). Version Control with Git and GitHub: Basic Git commands (clone, commit, push, pull, merge, branch), Working with Git repositories (local and remote), Collaboration using GitHub (pull requests, issue tracking). Configuration Management & Infrastructure as Code (IaC): Introduction to configuration management, Tools for automation (Ansible, Puppet, Chef, Terraform)
Case Studies: Set up and use Git for version control (clone, commit, push, pull, branching, merging).

Unit 5: Cloud Security & Compliance (7 Hours)

Security in Cloud: Shared Responsibility Model, Cloud Security Best Practices; Identity and Access Control: IAM Policies, MFA, Roles, Federated Access; Data Protection: Encryption at Rest/In Transit, Key Management (AWS KMS); Compliance & Auditing: AWS Artifact, CloudTrail, GDPR, HIPAA, PCI-DSS; Threat Detection: GuardDuty, AWS Inspector, Security Hub.
Case Study: Implement role-based access control in AWS with IAM & configure CloudTrail for activity logging

Unit 6: Emerging Trends (7 Hours)

How the Cloud Will Change Operating Systems, Location-Aware Applications, The Future of Cloud TV, Future of Cloud-Based Smart Devices, Faster Time to Market for Software Applications, Home-Based Cloud Computing, Mobile Cloud, Autonomic Cloud Engine, Multimedia Cloud, Energy Aware Cloud Computing, Jungle Computing, Industry-Specific Clouds, Advanced Cloud Security & Zero-Trust Models

List of Practical:

1. Setting Up a Cloud Account Create an account on a cloud provider (AWS, Azure, or Google Cloud) and explore the dashboard.
   I. Sign up for a Free Tier account (AWS, Azure, or GCP).
   II. Navigate the cloud provider's dashboard.

III. Set up billing alerts to avoid unexpected charges.
IV. Explore regions and availability zones.

2. Cloud Networking (VPC & Load Balancing)
   I. Create a Virtual Private Cloud (VPC).
   II. Set up a public and private subnet.
   III. Deploy two VMs in the VPC.
   IV. Set up a load balancer for traffic distribution.
3. Cloud Storage & File Uploads Store and retrieve files in the cloud.
   I. Create an S3 Bucket (AWS), Blob Storage (Azure), or Cloud Storage (GCP).
   II. Upload a file (image, document, etc.).
   III. Make the file publicly accessible (optional).Set up bucket permissions for security
4. Deploying a Static Website
   I. Upload an HTML page to S3 (AWS) or Azure Blob Storage.
   II. Enable static website hosting.
   III. Configure permissions to allow public access.
   IV. Access the website via a URL.
5. Cloud Security Basics (IAM & Permissions)
   I. Navigate to IAM (Identity & Access Management).
   II. Create a new user and assign roles & permissions.
   III. Test access with different permissions (Admin, Read-only, etc.).
   Set up MFA (Multi-Factor Authentication) for security.
6. Create a docker file with the following specs:
   I. Ubuntu container
   II. Apache2 installed
   III. Apache2 should automatically run once the container starts
7. Deploy a Kubernetes cluster for 3 nodes 2.
   I. Create a NGINX deployment of 3 replicas
   II. Create a service of type NodePort for NGINX deployment
   III. Check the NodePort service on a browser to verify
8. Deploying a Simple Cloud-Based Application
   I. Launch a web server VM.
   II. Store website assets in cloud storage.
   III. Set up a database (RDS, Cloud SQL, or CosmosDB).
   IV. Configure DNS & Load Balancer for high availability

List of Project areas:

1. Cloud-Based Web Application Deployment with CI/CD
2. Auto-Scaling Microservices Architecture using Containers
3. Serverless Application Development and Deployment
4. Infrastructure as Code (IaC) for Automated Cloud Provisioning
5. Cloud Security Implementation and Monitoring System
6. DevOps Pipeline for Continuous Testing and Deployment
7. Cloud-Based Data Backup and Disaster Recovery System

Text Books :

1. Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley,India.
2. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India
3. Gautam Shroff. "Enterprise Cloud Computing", Cambridge

Reference Books:

1. Cloud Computing: Concepts, Technology & Architecture Authors: Thomas Erl, Zaigham Mahmood, and Ricardo Puttini.
2. Architecting the Cloud: Design Decisions for Cloud Computing Service Models Author: Michael J. Kavis
3. Barrie Sosinsky, "Cloud Computing Bible", Wiley India

e Learning Resources:

1. https://online-pmo.com/wp-content/Education/Learning%20DevOps.pdf
2. https://git-scm.com/book/en/v
3. https://www.techmagic.co/blog/best-application-deployment-strategies
4. https://www.jenkins.io/doc/

Course Outcomes:

Students will be able to:

CO1: Illustrate the benefits of cloud computing and its applications with respect to various service models.

CO2: Apply virtual machine concepts and establish networking.

CO3: Evaluate and select appropriate storage solutions (object, block, file) based on performance, cost, and scalability needs.

CO4: Design and implement CI/CD pipelines using tools such as Jenkins or GitHub Actions for automated build, test, and deployment processes.

CO5: Design and implement identity and access management solutions using IAM policies, roles, MFA, and federated access.

CO6: Explore the emerging trends in the Cloud computing environment.

CO-PO Mapping Matrix: Object Oriented Programming

CO	Program Outcomes (PO)												PSO	
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	-	-	-	-	1	-	-	-	-	2	2	1
CO2	3	2	3	-	3	-	-	-	-	-	-	-	3	1
CO3	-	3	3	2	3	-	-	-	-	-	2	-	2	1
CO4.	-	-	3	2	3	-	-	-	2	2	2	-	2	1
CO5	-	3	3	-	3	2	-	2	-	-	-	-	1	3
CO6	2	-	-	-	3	-	1	-	-	-	-	3	2	2
Average	2.0	2.5	3.0	2.0	2.8	2.0	1.0	2.0	2.0	2.0	2.0	2.5	2.0	1.5

Scale:

3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

CO1: Illustrate the benefits of cloud computing and its applications with respect to various service models.

- **PO1 (3):** Requires understanding of cloud computing fundamentals and service models (IaaS, PaaS, SaaS).
- **PO2 (2):** Involves analyzing cloud solutions for different applications.
- **PO7 (2):** Cloud computing promotes efficient resource utilization and sustainability.
- **PO12 (2):** Encourages continuous learning of evolving cloud technologies and services.
- **PSO1 (2):** Cloud platforms are essential for IoT data storage and processing.
- **PSO2 (1):** Basic understanding of cloud supports secure system awareness.

CO2: Apply virtual machine concepts and establish networking.

- **PO1 (3):** Uses knowledge of virtualization and networking concepts.
- **PO2 (2):** Analyzes networking requirements and VM configurations.
- **PO3 (2):** Designs virtualized computing environments.
- **PO5 (3):** Utilizes modern cloud and networking tools for deployment and management.
- **PSO1 (3):** Strongly supports IoT system deployment (device connectivity, cloud integration).
- **PSO2 (1):** Provides basic infrastructure knowledge for secure systems.

CO3: Evaluate and select appropriate storage solutions (object, block, file) based on performance, cost, and scalability needs.

- **PO2 (3):** Analyzes storage requirements and constraints
- **PO3 (2):** Designs suitable storage architectures
- **PO4 (2):** Evaluates storage solutions through performance analysis and benchmarking.
- **PO5 (2):** Applies cloud storage technologies and tools.
- **PO11 (1):** Considers cost optimization and resource management in storage selection.
- **PSO1 (2):** IoT systems generate large data → requires efficient storage selection.
- **PSO2 (1):** Limited relation to blockchain storage concepts.

CO4: Design and implement CI/CD pipelines using tools such as Jenkins or GitHub Actions for automated build, test, and deployment processes.

- **PO3 (3):** Designs automated software delivery workflows.
- **PO4 (2):** Investigates and validates deployment strategies.
- **PO5 (3):** Uses modern DevOps tools and platforms..
- **PO9 (2):** Supports collaborative software development.
- **PO10 (2):** Requires documentation and communication of deployment processes.
- **PO11 (2):** Involves project planning, resource management, and automation practices.
- **PSO1 (2):** Helps in continuous deployment of IoT applications.

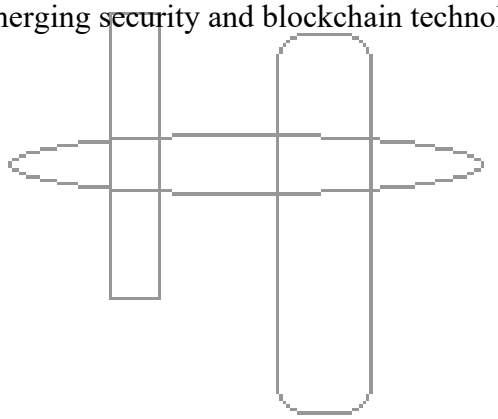
- **PSO2 (1):** Minor role in automating secure/blockchain-based systems.

CO5: Design and implement identity and access management solutions using IAM policies, roles, MFA, and federated access.

- **PO2 (3):** Analyzes security requirements and risks.
- **PO3 (2):** Designs secure access control mechanisms.
- **PO5 (2):** Uses cloud security and IAM tools..
- **PO6 (2):** Addresses legal, privacy, and security concerns in organizations.
- **PO8 (3):** Emphasizes ethical handling of user identities and access privileges..
- **PSO1 (1):** Limited direct role in IoT system design.
- **PSO2 (3):** Strongly aligned with security, authentication, authorization, and secure system design.

CO6: Explore the emerging trends in the Cloud computing environment.

- **PO1 (3):** Strong emphasis on lifelong learning and adaptability.
- **PO5 (2):** Analyzing new technologies and trends.
- **PO7 (2):** Understanding societal/environmental impact.
- **PO12 (1):** Basic investigation of new technologies.
- **PSO1 (2):** Includes IoT-cloud integration trends (Edge, Fog computing).
- **PSO2 (2):** Covers emerging security and blockchain technologies.



FF No. : 654

CB3203C: Human Computer Interaction

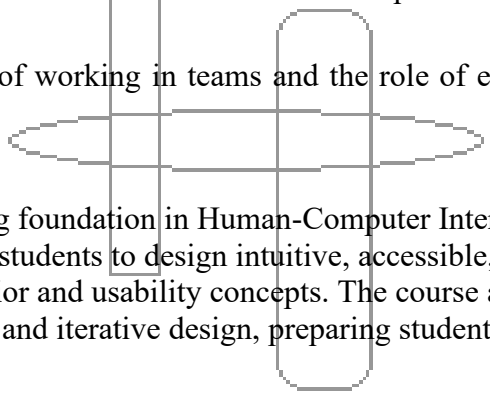
Credits: 4 **Teaching Scheme:** Theory: 3 Hours/Week, Lab: 2 Hours/Week

Prerequisites

1. Software Project Management

Course Objectives

1. Provide an overview of the concepts relating to the design of human-computer interfaces in ways making computer-based systems comprehensive, friendly and usable. To Learn different protocols used for IoT design .
2. Understand the theoretical dimensions of human factors involved in the acceptance of computer interfaces.
3. Understand the important aspects of implementation of human-computer interface
4. Identify the various tools and techniques for interface analysis, design, and evaluation.
5. Identify the impact of usable interfaces in the acceptance and performance utilization of information systems.
6. Identify the importance of working in teams and the role of each member within an interface development phase.



Course Relevance

This course provides a strong foundation in Human-Computer Interaction (HCI) and user-centered design principles. It enables students to design intuitive, accessible, and effective user interfaces by understanding human behavior and usability concepts. The course also develops skills in interface evaluation, usability testing, and iterative design, preparing students to create user-friendly digital solutions.

SECTION I

Unit 1: Introduction to Human-Computer Interaction (7 Hours)

Includes the difference between good and poor interaction design, what interaction design is and how it relates to human-computer interaction and other fields, what is involved in the process of interaction design, the different forms of guidance used in interaction design, etc. Need for Design Examples from Design of everyday things, case studies, Evolution of the web and digital interfaces, Design thinking and wicked problems.

Exercise - Identify problems around us requiring design solutions Or problems solved using design.

Unit 2: Understanding the Design Process (7 Hours)

Interaction design basics, HCI in the software process, Design rules, Implementation support, Evaluation techniques, Universal design, User support, Individual differences, designing interfaces for all, User research and techniques, Understanding Personae, Good and poor design, Ergonomics.

Exercise - Creating personae for different applications in everyday use.

Unit 3: Understanding the Interaction (7 Hours)

Understanding device specific interactions and human aspects involved, Interaction styles, interacting with voice, visual and audio-visual interfaces, Understanding Scenarios and context of use for user, Understanding user journey and user journey maps. Interaction Styles- Direct Manipulation and Virtual Environments, Menu Selection, Form Filling and Dialog Boxes, Command and Natural Languages, Interaction Devices, Collaboration and Social Media Participation

Exercise - Listing down scenarios for an application/system, critical task list for each scenario

Exercise -Creating a user journey map for a particular task

SECTION II

Unit 4: Design Issues (7 Hours)

Introduction to UX design process and case study, Understanding Information Architecture, Overview of tools. Understanding navigation models based on information architecture, High level concept sketches/wireframes. Quality of Service, Balancing Function and Fashion, User Documentation and Online Help, Information Search, Information Visualization.

Exercise - Open and closed card sorting technique - Creating information architecture for a system

Exercise - Creating low fidelity concept sketches for critical tasks of system/problems.

Unit 5: UI Evaluation Techniques (7 Hours)

What, why and when to evaluate, Design guidelines, Golden rules and heuristics, Goals of Evaluation, Evaluation criteria, Evaluation through: Expert analysis, User participation, Testing techniques – Formative and Summative testing, surveys, peer reviews and so on.

Case study - ROI on UX/HCI methodology

Unit 6: Cognitive Models (7 Hours)

Cognitive models, Goal and Task hierarchy models, Linguistic models, Physical and Device models, Design principles. Introduction to Prototyping tools, Understanding how UI Interaction & Prototype Design works, UX -Industry overview. Prototyping / wire framing tool exercise: Design a small mobile app design, along with user journeys and multiple interaction touch points using Figma or Balsamiq.

Exercise - Conduct evaluation of different sample interfaces using different models.

List of Practical:

1. Design of Everyday Things: Find 3 unique usability problems in the day-to-day objects that you have observed or encountered. Describe them in 3 slides in a presentation.

2. User Research- Contextual Inquiry, interview (planned with proper questions) and feedback: In groups of two, complete user research for a common real-life application or system by conducting a contextual inquiry and structured interviews. Choose one of the following topics.
   - Wayfinding experience for new students in your college campus
   - Online food delivery experience for hostel students
   - Experience of using a library search/catalog system
   - Accessing public transport or bus timetable apps

- Booking train tickets on IRCTC or similar
- Digital payment apps usage by small vendors
- Online learning platform experience (e.g., Coursera, Moodle)

3. User Persona: 1).Creating personae for different application in everyday use (Continue same topic as assign 2). Develop a precise description of the user of the product in focus of the design and what the user wishes to accomplish.

4. Card Sorting: Conduct open and closed card sorting to define Information Architecture of a library or movie rental website.

5. User journey and User journey maps: Listing down scenarios for an application/system, critical task list for each scenario. Creating a user journey map for a particular task.

6. Low-Fidelity Wireframing: Design low-fidelity sketches for a to-do list or bus booking app. Tools: Paper sketching or Balsamiq.

7. Figma Prototype : Create a prototype for a chosen app scenario and conduct formative usability testing with 3 users.

8. Usability Testing: usability test feedback, and redesign suggestions.

9. Analyze different interaction styles (menu-based, command-line, direct manipulation, etc.) in existing systems and evaluate their effectiveness for different user tasks.

10. Project: Design a Solution for a Social Problem through HCI

List of Project areas:

- UX Design and Prototype of a Smart Health Monitoring Mobile App
- Redesign of a Food Delivery Mobile App for Improved Usability
- Development of a User-Friendly Digital Library Web Interface
- Usability Testing and Improvement of a Government Service Website
- Voice-Based Navigation System for Inclusive Interaction
- Design of an Interactive Public Transport Information System
- Navigation Model Design for a Learning Management System
- Persona Creation and User Journey Mapping for a Food Delivery App
- Visualization of Healthcare Data Using User-Friendly Interfaces

Text Books:

1. “Human Computer Interaction” by Alan Dix, Janet Finlay, ISBN :9788131717035, Pearson Education (2004)

2. “Designing the User Interface - Strategies for Effective Human Computer Interaction”, by Ben Schneiderman, ISBN: 9788131732557, Pearson Education (2010).

Reference Books:

1. Usability Engineering: Scenario-Based Development of Human- Computer Interaction, by Rosson, M. and Carroll, J. (2002)
2. The Essentials of Interaction Design, by Cooper, et al., Wiley Publishing (2007)
3. Usability Engineering, by Nielsen, J. Morgan Kaufmann, San Francisco, 1993. ISBN 0-12-518406-9
4. The Resonant Interface: HCI Foundations for Interaction Design, by Heim, S., Addison-Wesley. (2007)
5. Usability engineering: scenario-based development of human- computer interaction, By Rosson, M.B & Carroll, J.M. , Morgan Kaufman.(2002)

e Learning Resources:

1. Enhancing the customer experience with Human Computer Interaction (Infosys springboard)
2. Artificial Intelligence: Human Computer Interaction (Infosys Springboard)
3. Human-Computer Interaction by edX (offered by Georgia Tech)
4. Human-Computer Interaction by Udacity

Course Outcomes:

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

CO1: Understand the Human-Computer Interaction and what interaction design is.
CO2: Understand Human Capabilities and Core Cognitive aspects of interaction design.
CO3: Design Interactive Systems with clear understanding of HCI principles that influence a system's interface design, before writing any code.
CO4: Create interfaces for different environments, people, places, and activities.
CO5: Apply Quantitative Analysis, Evaluation, and Redesign.
CO6: Evaluate sample interfaces using different models.

CO-PO Mapping Matrix:

CO	Program Outcomes (PO)													
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PO 12	PSO1	PSO2
CO1	3	2	-	-	2	3	2	-	-	-	-	2	2	-
CO2	3	3	3	2	3	-	-	-	2	-	-	-	3	2
CO3	2	3	2	3	2	-	-	2	2	-	-	-	2	2
CO4	2	2	3	2	3	-	-	2	3	1	-	1	3	3
CO5	2	3	2	3	2	2	-	3	3	-	-	2	2	3
CO6	1	2	3	2	3	1	-	2	2	-	-	3	2	3
Average	2.17	2.5	2.17	2.0	2.5	2.0	2.0	2.2	2.5	1.0	-	2.0	2.33	2.5

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification for Mapping

CO1: Understand the Human-Computer Interaction and what interaction design is.

- **PO1 (3):** Strong foundation of HCI concepts and principles.
- **PO2 (2):** Helps in identifying and understanding interaction-related problems.
- **PO10 (1):** Basic communication of design concepts.
- **PO12 (2):** Supports continuous learning in evolving HCI field.

CO2: Understand Human Capabilities and Core Cognitive aspects of interaction design.

- **PO1 (3):** Applies knowledge of human cognition and perception.
- **PO2 (3):** Strongly used in analyzing user behavior and limitations.
- **PO3 (2):** Helps in designing cognitively suitable interfaces.
- **PO10 (2):** Communicates user-centered insights.
- **PO12 (2):** Keeps updated with cognitive and usability research.

CO3: Design Interactive Systems with clear understanding of HCI principles that influence a system's interface design, before writing any code.

- **PO2 (2):** Identifies user requirements and constraints.
- **PO3 (3):** Core focus on designing interactive systems.
- **PO4 (2):** Involves validation through testing and analysis.
- **PO5 (2):** Uses modern tools for design and prototyping.
- **PO10 (2):** Communicates design solutions effectively.
- **PO12 (2):** Encourages adaptation to new design tools.

CO4: Create interfaces for different environments, people, places, and activities.

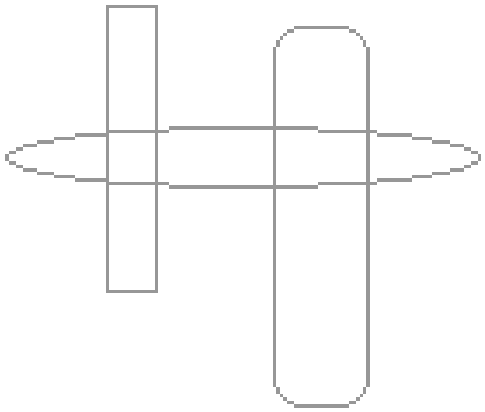
- **PO2 (2):** Analyzes diverse user needs and contexts.
- **PO3 (3):** Designs inclusive and adaptive interfaces.
- **PO4 (2):** Evaluates usability in different environments.
- **PO5 (2):** Uses tools for multi-platform interface design.
- **PO9 (1):** Some teamwork involved in design tasks.
- **PO10 (2):** Communicates interface design effectively.

CO5: **Apply Quantitative Analysis, Evaluation, and Redesign.**

- **PO2 (3):** Strong focus on analyzing usability data.
- **PO3 (2):** Improves designs based on evaluation.
- **PO4 (3):** Conducts testing, surveys, and analysis.
- **PO5 (2):** Uses evaluation and analytics tools.
- **PO10 (2):** Reports evaluation findings clearly.
- **PO12 (2):** Supports iterative learning and improvement.

CO6: Evaluate sample interfaces using different models.

- **PO2 (3):** Strong analytical evaluation of interfaces.
- **PO3 (2):** Suggests improvements based on evaluation.
- **PO4 (3):** Applies structured evaluation techniques.
- **PO5 (2):** Uses modern evaluation tools.
- **PO10 (2):** Communicates evaluation results.
- **PO11 (1):** Minor project/resource consideration.
- **PO12 (3):** Strong emphasis on continuous learning of UX trends.



FF No. : 654

ID3001: Basics of Game Development

Credits: 4 **Teaching Scheme:** Theory: 3 hours/Week, Lab: 2 Hours/Week

Prerequisites

1. Knowledge of OOP
2. Basic Java Script

Course Objectives

1. Learn Game Mechanics and web based game development concepts.
2. Learn C# OOP fundamentals in order to do Unity 3D scripting.
3. Learn programming patterns that are generally used in game development.
4. Learn the basics of game services.
5. Learn the basics of collision physics in gaming.

Course Relevance

This course introduces the fundamentals of modern game development, including game design, programming, game physics, and networking concepts. It enables students to develop interactive games using web-based and standalone platforms with industry-standard tools. The course also builds practical skills in designing, testing, and deploying games while following software engineering best practices.

SECTION I

Unit 1: Game Domain **(7 hours)**

Gaming as business & Global market, Introduction to Gaming, Gaming Ecosystem and opportunities, Player time spending, what kind of games succeed, monetizing game, how to target the gaming crowd, what is working for real money gaming, game testing, types of game testing, Realism and compliance testing, multiplayer and network testing.

Unit 2: Web based Game Programming **(7 hours)**

Basics of HTML, CSS, Java Script, introduction to phaser.io js library

Unit 3: Standalone Game Platform **(7 hours)**

C# fundamentals with all OOPs concepts, Introduction to Unity 3D Engine, C# scripting in Unity 3D Engine

SECTION II

Unit 4: Programming Design Patterns in Game Development (7 hours)

Observer Design Pattern, State Design Pattern, Command Design Pattern, Decorator Design Pattern

Unit 5: Game Services (7 hours)

Choosing a Gamer Service, Basic Setup, Lobbies and Match making, Networking, Player Statistics, Player Achievements Leaderboards, Other Service

Unit 6: Game Physics: Collision Detection (7 hours)

The Collision Detection Pipeline, Broad Phase Collision Detection, Requirements, Bounding Volume Hierarchies, Hierarchies Building the Hierarchy, Sub-object Hierarchy, Spatial Partitioning, Binary Space Partitioning, Oct-Trees and Quad-Trees, Grids, Multiresolution Maps

List of Practical:

1. Snake Game

Develop a simple Snake game using JavaScript and H5 and a game framework such as Phaser.io. The game should include the following features:
1. A snake that can move in four directions (up, down, left, right) using the arrow keys.
2. Food items that randomly appear on the game field.
3. The snake grows longer each time it eats a food item.
4. The game ends when the snake collides with itself or the walls.
5. A scoring system that tracks the number of food items eaten.

2. Tetris Game

Develop a simple Tetris game using C# and a game framework such as Unity. The game should include the following features:
1. Randomly generated Tetriminos (tetrominoes).
2. The ability to move and rotate Tetriminos.
3. The game field (grid) should be able to detect filled rows and clear them.
4. The game should end when the Tetriminos reach the top of the grid.
5. A scoring system that tracks the number of cleared lines.

3. State Machines in Gaming

Implementation Related to State Machines: Implement a state machine to manage the state of a traffic light system. Write the code for a traffic light state machine with states like “Green,” “Yellow", "Red” and transitions between these states.

How would you handle asynchronous actions within a state machine?

Provide examples of handling asynchronous actions (like API calls) within a state machine using JavaScript.

4. Scenario Based statement

Describe a scenario where you would use JavaScript for UI and C# for game logic in a game. Provide an example where this separation of concerns is beneficial and explain how the integration would work.

How would you debug issues in a game with integrated JavaScript and C# components? Outline the tools and techniques you would use to debug problems that arise from the interaction between JavaScript and

How would you implement cross-platform support for a game using JavaScript and C#? Describe strategies for ensuring that a game runs smoothly on multiple platforms (e.g., web, mobile, desktop).

5. Core Gaming (Understanding, Analysis and Presentation)

Flowchart/Presentation Title: Enhancing Player Engagement in Candy Crush/Temple Run/Puzzle game Using Design Patterns

Observer Pattern
Description: Real-time notifications for friend activities and game events.
Impact: Increases social interaction and competitive spirit.
Visual: Diagram showing notification flow from game server to players.

State Pattern
Description: Manage different game states for varied gameplay.
Impact: Keeps the game fresh and challenging.
Visual: Flowchart showing transitions between Normal Play, Special Event, and Timed Challenge states.

Command Pattern
Description: Allow players to undo moves and activate abilities.
Impact: Enhances control and reduces frustration.
Visual: Sequence diagram illustrating the Command pattern for undoing a move.

Decorator Pattern
Description: Add dynamic features like power-ups and skins.
Impact: Provides customization and boosts player excitement.
Visual: Diagram showing how decorators add new features to game components.

Conclusion
Summary: Recap of the suggested patterns and their benefits. Next Steps: Implementation plan and expected outcomes.

6. Async and sync Implementation in Gaming

How do you define an asynchronous function in JavaScript? Provide examples of defining asynchronous functions using `async` and `await` keywords.

What are Promises and how are they used in JavaScript? Explain the concept of Promises and their role in managing asynchronous operations.

Implement a basic game loop using synchronous functions in JavaScript. Write code for a simple game loop that updates game state and renders frames synchronously.

Convert the above synchronous game loop into an asynchronous one. Rewrite the game loop using `requestAnimationFrame` and asynchronous functions to prevent blocking the main thread.

How would you handle asynchronous loading of game assets (e.g., images, sounds) in JavaScript? Provide code examples for loading game assets asynchronously using Promises or async/await.

Implement a function to fetch high scores from a server asynchronously in a JavaScript game Write a function that uses `fetch` or `XMLHttpRequest` to retrieve high scores from an external server without blocking the game.

Describe how you would manage asynchronous player inputs in a multiplayer game Explain techniques for handling player inputs asynchronously, ensuring that the game remains responsive and updates in real-time.

Reference Books:

1. Colt McAnlis, et al., "HTML5 Game Development Insights", 2014, Apress, ISBN: 978-1-4302-6698-3. (https://doi.org/10.1007/978-1-4302-6698-3)
2. Joshua Glazer, Sanjay Madhav, " Multiplayer Game Programming, Architecting Networked Games", 2016, Addison-Wesley, ISBN-13: 978-013-403430-0
3. Ian Millington, "Game Physics Engine Development", 2007, Morgan Kaufmann, ISBN-13: 978-0-12-369471-3

e learning Resources:

1. Udemy
2. Coursera

Course Outcomes:

After completion of the course, students will be able to:
CO1: Explain the fundamentals of the gaming industry, game development lifecycle, gaming platforms, and business models.
CO2: build simple interactive games using appropriate game development frameworks and programming languages for web and standalone platforms.
CO3: Apply game programming concepts, software design patterns, and state management techniques to develop maintainable game applications
CO4: Apply essential game development features such as physics, collision detection, game services, networking, and asynchronous programming.
CO5: Analyze game architectures, programming approaches, and implementation strategies for solving real-world game development scenarios and enhancing player experience.

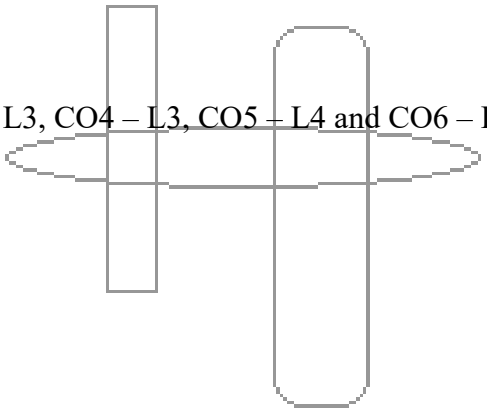
CO6: Design, develop, and present game-based solutions by integrating appropriate game development tools, programming techniques, and industry practices.

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	-	3	1	-	-	-	-	-	-	3	1	-	-
CO2	3	2	3	-	3	1	-	-	2	-	-	-	3	2	-	-
CO3	3	2	3	-	3	1	-	-	2	-	-	-	3	3	2	-
CO4	3	2	3	-	3	2	1	1	1	-	1	-	3	2	-	-
CO5	3	2	3	-	3	2	1	2	2	-	2	2	3	1	1	1
CO6	3	2	3	-	3	3	1	3	3	3	3	3	3	1	-	2
Average	**3**	**2**	**3**	**0**	**3**	**1.67**	**2**	**2**	**2**	**3**	**2**	**2.5**	**3**	**1.67**	**1.5**	**1.5**

CO Attainment levels:

Weights for attainment levels:
L1 - Easy-0.75
L2 - Comfortable-0.7
L3 - Medium-0.65
L4 - Somewhat difficult-0.6
L5 - Difficult-0.55
CO1 – L1, CO2– L2, CO3 – L3, CO4 – L3, CO5 – L4 and CO6 – L4



FF No. : 654

ID3002: Mainframe Technique

Credits: 4 **Teaching Scheme:** Theory: 3 Hours/Week, Lab: 2 Hours/Week

Prerequisites

1. Problem Solving Skills

Course Objectives

1. To introduce IBM Z Mainframe Technology.
2. To learn industry-critical enterprise computing skills.
3. To introduce IBM Mainframe Operating System z/OS.
4. To learn z/OS Access and Interaction Basics.
5. To study the Operating System Environment for computing.
6. To learn z/OS REXX, COBOL Programming

Course Relevance

A mainframe technology course remains highly relevant for building a stable, high-paying IT career. Enterprises in banking, insurance, and government rely on mainframe systems for mission-critical operations, massive transaction volumes, and high-security needs, creating steady demand for skilled professionals to maintain and modernize these systems.

SECTION I

Unit 1: Introduction to IBM Z Mainframe Technology (4 Hours)

Introducing IBM Z Hardware, IBM z16, IBM z16 single frame, IBM z16 rack mount, IBM z15 T01 and T02 machines, Applications of IBM Z, Modern Mainframe innovations: z15 and z16 systems, Containerization and Kubernetes on Mainframes, Mainframe Hardware components such as Central Processing Unit (CPU), Memory and Storage Systems, Input/Output Devices, Case studies of Mainframe implementations.

Unit 2: Learn Industry-Critical Enterprise Computing Skills (4 Hours)

IBM Z Mainframe Skills Depot: Mainframe System Administrator, Mainframe Application Developer/Programmer, Mainframe Modernization Architect, Database Administrator (DBA), Storage Administrator, System Programmer (SysProg), Network Administrator, Mainframe DevOps Engineer, Cybersecurity Engineers, to name a few job roles in the vast and amazing world of the IBM Z Mainframes.

Unit 3: z/OS Introduction (6 Hours)

z/OS: Essential Technology of Large Enterprises, Introducing IBM Mainframe Operating Systems, z/OS - IBM Z Flagship Operating System, z/OS Professional Documentation, Architecture of IBM Z systems, networking, and security aspects.

SECTION II

Unit 4: z/OS Access and Interaction Basics (7 Hours)

Time Sharing Option, Interactive System Productivity Facility (ISPF) Editor, System Display and Search Facility (SDSF), TN3270, SSH, FTP. Introduction to Structured Programming Datasets and Data Access Methods: Dataset Names and Types, Sequential Dataset, Partitioned Datasets, VSAM - Virtual Storage Access Method, Disk Storage, Volume Labels, VTOC, Extents, System Utilities and Tools, Dataset management utilities (IEBCOPY, IEFBR14, IEBGENER, IDCAMS, etc.), DFSORT basic and examples

Unit 5: Operating System Environment (7 Hours)

Unix System Services (USS), JCL - Job Control Language, JES - Job Entry Subsystem, System Catalog, Components, Messages, System Log, Virtual Storage and Address Spaces, System and User Address Space Management, Key Controlled Protection for Address Spaces, Security, RACF, Communications Server, z/OS Operator Commands, z/OS Initial Load Program Process, Virtualization and Cloud computing. Data Definition (DD) Statement: Connecting a physical dataset to a program's internal logical filename. SPACE allocation (Primary, Secondary, Directory blocks for PDS, and RLSE).

Unit 6: Mainframe Programming (8 Hours)

z/OS REXX programming, Learning COBOL programming with VS code, z/OS Unix System Services (USS), JCL (Job Control Language), Interactive System Productivity Facility (ISPF), Java on z/OS, Virtual Storage Access Method (VSAM), Ansible Automation, creating fully automated CI/CD pipelines involving components residing on z/OS, Zowe, Big Data and Analytics: Hadoop on Mainframes, Machine Learning Integrations. Mainframe Application Lifecycle Management. Agile project Methodology.

Level Numbers: Structural levels (01-49), elementary items (77), condition names (88). Computational data types like Packed Decimal (COMP-3). Looping/Iteration: PERFORM, UNTIL, TIMES, USING (Parameters), Perform Through , REDEFINES and RENAMES clauses, Arrey (single and double Dimension), Static and Dynamic Table in Working Storage.

List of Practical

Job Control Language (JCL) Assignments:

1) Dataset operations (Create, Delete, Rename, Copy, Catalog, Uncatalog, etc.)
2) Submit Job, Redirect Output, Check Job Status and Output.
3) Condition code handling, JCL overrides.
4) Creating an In-stream PROC with symbolic parameters, calling it multiple times based on return codes.

COBOL Programming Assignments:

1) Control Structures: If-Else Condition and Simple Loop (PERFORM UNTIL)
2) File Handling: Sequential and Indexed (VSAM)
3) Perform varying, Error handling, etc.
4) Create a file using Data definitions using precise byte alignments for personal and business customers / accounts.

For MOOCs and other learning Resources

Unit-1:
- https://www.ibm.com/z/education
- https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html

Unit-2:
- https://www.youtube.com/watch?v=aEXHOBYv56E

Unit-3:
- https://www.youtube.com/watch?v=aEXHOBYv56E
- https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html

Unit-4:
- https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html

Unit-5:
- https://learn.ibm.com/course/view.php?id=9890
- https://publibz.boulder.ibm.com/zoslib/books/tutorials/introjcl/index.html

Unit-6:
- https://openmainframeproject.org/projects/cobol-programming-course/
- https://learn.ibm.com/course/view.php?id=9890
- https://learn.ibm.com/course/view.php?id=7552
- https://learn.ibm.com/course/view.php?id=8722
- https://publibz.boulder.ibm.com/zoslib/books/tutorials/introjcl/index.html

E-learning Resources:

- https://ibmzxplore.influitive.com/users/sign_in

- https://www.ibm.com/z/education
- https://www.tutorialspoint.com/cobol/cobol_overview.htm

MOOC Courses:

1. https://www.ibm.com/training/
2. https://learn.ibm.com/
3. https://www.ibm.com/z/education
4. https://www.redbooks.ibm.com/redbooks.nsf/redbookabstracts/crse0304.html

Course Outcomes:

After completion of the course, students will be able to:
CO1: Understanding of IBM Z Mainframe Technology.
CO2: Apply the industry-critical enterprise computing skills.
CO3: Apply IBM Mainframe Operating System z/OS for Large Enterprise.
CO4: Apply z/Os Access and Interaction Basics.
CO5: Design and develop a program for Job Control Language using z/OS.
CO6: Design and develop a program for REXX, COBOL using z/OS.

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	1	2							2	2	3		
CO2	3	3	2	1	3				1			2	3	3	2	
CO3	3	3	2	2	3							2	3	3	2	1
CO4	2	3	2	2	3				1	1		2	3	3	2	1
CO5	3	3	3	2	3				1	1	1	2	3	3	3	2
CO6	3	3	3	2	3				2	2	1	3	3	3	3	2
Average	2.83	2.83	2.17	1.67	2.83				1.25	1.33	1.00	2.17	2.83	3.00	2.40	1.50

CO Attainment levels:

Weights for attainment levels:

L1 - Easy-0.75
L2 - Comfortatble-0.7
L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6
L5 – Difficult – 0.55

CO1 – L3, CO2– L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

FF No. : 654

MM0501A: Operating Systems and Distributed Systems

Credits: 3 **Teaching Scheme:** Theory: 2 hours/Week, Tutorial:1Hour/Week

Prerequisites

1. Fundamentals of Computer Organization
2. Basics of Programming (C/Python/Java)
3. Data Structures and Algorithms

Course Objectives

1. To introduce key principles of operating systems and their role in computer systems.
2. To explore process management, CPU scheduling, and synchronization techniques.
3. To analyze memory and file management strategies employed in OS.
4. To understand the architecture, communication models, and coordination in distributed systems.
5. To explore real-world distributed file systems and synchronization protocols.
6. To develop practical understanding through implementation and simulation.

Course Relevance

This course provides a strong foundation in Operating Systems and Distributed Systems, covering process, memory, file, and resource management. It enables students to understand and optimize the performance of modern operating systems such as Linux, Windows, and Android. The course also introduces distributed computing concepts, preparing students for careers in system software, cloud computing, and IT infrastructure.

SECTION I

Unit 1: Introduction to Operating Systems (7 Hours)

Operating system objectives and functions, Evolution of operating systems, System structures, Operating system services, User and operating system interface, System calls, Types of operating systems, Process concepts, Process scheduling, Operations on processes, Inter-process communication, Multithreading models.

Case Study - Android vs Linux vs Windows OS Architecture – Comparative Study

Tool : Use of QEMU / Docker to illustrate different OS kernels

Unit 2: Process Management and Synchronization (7 Hours)

CPU scheduling algorithms, Thread scheduling, Synchronization tools, Critical-section problem, Semaphores, Monitors, Classical problems of synchronization, Deadlocks: system model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

Use case- Dining Philosophers in POSIX Threads or Go Channels

Case Study: Deadlocks in Database Systems / Cloud Containers (e.g., Kubernetes)

SECTION II

Unit 3: Memory Management and File Systems (7 Hours)

Memory management strategies, Swapping, Contiguous memory allocation, Paging, Structure of the page table, Segmentation, Virtual memory, Demand paging, Page replacement algorithms, Allocation of frames, Thrashing, File system interface, File concept, Access methods, Directory structure, File system mounting, File sharing, Protection.
Case Study: Virtual Memory in Windows and Linux – Swap vs Pagefile
Tool: Use top, htop, and vmstat in Linux for live memory and file I/O monitoring

Unit 4: Introduction to Distributed Systems (7 Hours)

Definition and goals of distributed systems, Hardware and software concepts, Design issues, Communication in distributed systems, Remote procedure calls, Remote method invocation, Process synchronization in distributed systems, Distributed file systems, Distributed coordination, Case studies: Hadoop Distributed File System (HDFS), Google File System (GFS).
Case Studies: 'Dropbox Sync Engine', 'Amazon DynamoDB', 'Blockchain consensus as distributed coordination'
Tool: Simulate Remote Procedure Call using gRPC (in Python or Go)

List of Practical's:

1. Study and installation of a Linux-based OS (Ubuntu/Fedora) and basic shell usage
2. Exploring Android Emulator Internals to understand mobile OS structures
3. Simulation of CPU scheduling algorithms (FCFS, SJF, RR, Priority)
4. Implementation of multithreading and synchronization (Pthreads or Java Threads) / Synchronization problems (Readers-Writers, Producer-Consumer) using POSIX threads or Go routines
5. Solve producer-consumer problem using semaphores / Solve readers-writers problem using monitors
6. Simulate deadlock avoidance using Banker's algorithm / Use Docker containers to demonstrate process isolation and deadlock simulation
7. Simulate page replacement algorithms (FIFO, LRU, Optimal)
8. File system operations using C/Java/Linux shell (file creation, access, directory ops)
9. Implementation of interprocess communication using sockets or shared memory
10. Use Wireshark to trace distributed file system calls or RPC calls
11. Implementation of Remote Procedure Call using Java RMI or gRPC
12. Simulate logical clocks (Lamport) and basic distributed mutual exclusion
13. Case study implementation or simulation of HDFS or a simple distributed file system
14. Distributed mutual exclusion using Ricart-Agrawala or Lamport timestamps in Python Mini-project:
15. Build a toy distributed file sync service (like rsync or Dropbox Lite)

List of Project Areas:

1. Implement a basic RPC with gRPC in Python for two services to communicate.
2. Simulate logical clock synchronization across processes.
3. Build a simple demand paging simulator showing page faults and replacement.

Text Books:

1. A. Silberschatz, P. Galvin, G. Gagne; Operating System Concepts; 10th Ed.; Wiley; 2018
2. A. S. Tanenbaum, M. van Steen; Distributed Systems: Principles and Paradigms; 2nd Ed.; Pearson; 2007

Reference Books:

1. W. Stallings; Operating Systems: Internals and Design Principles; 9th Ed.; Pearson; 2018
2. G. Coulouris, J. Dollimore, T. Kindberg; Distributed Systems: Concepts and Design; 5th Ed.; Pearson; 2012
3. D. Bovet, M. Cesati; Understanding the Linux Kernel; 3rd Ed.; O'Reilly; 2005
4. B. Burns, B. Grant; "Designing Distributed Systems"; O'Reilly; 2018; [Free PDF from Microsoft: https://azure.microsoft.com/en-us/resources/designing-distributed-systems/]
5. M. Tim Jones; "Linux Kernel Development – A Practical Approach"; IBM DeveloperWorks; Open Resource

e-Learning Resources:

1. Advanced Distributed Systems: https://onlinecourses.nptel.ac.in/noc26_cs169/preview?
2. Introduction To Operating Systems: https://onlinecourses.nptel.ac.in/noc26_cs123/preview?
3. Programming Fundamentals : Command Line Interface & Operating System Commands - TOC | Infosys Springboard

Course Outcomes:

After completion of the course, student will be able to

CO1: Explain operating system concepts including structure, services, process management, and compare architectures of Android, Linux, and Windows.

CO2: Apply process scheduling and synchronization techniques to solve critical section problems and analyze deadlock handling methods.

CO3: Analyze memory management techniques and file system structures, and evaluate system performance using appropriate tools.

CO4: Explain distributed system concepts and implement communication and coordination mechanisms using technologies like RPC and gRPC.

CO-PO Mapping Matrix:

COs	Program Outcomes (POs)												PSO	
COs \ POs/PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	0	1	0	0	0	0	1	0	1	2	1

CO2	3	3	2	1	2	0	0	0	1	0	0	1	2	2
CO3	2	3	2	2	3	0	0	0	0	1	0	1	2	2
CO4	2	2	3	2	3	1	0	0	1	2	1	1	3	2
Average	**2.5**	**2.5**	**2.0**	**1.25**	**2.25**	**0.25**	**0**	**0**	**0.5**	**1.00**	**0.25**	**1.00**	**2.25**	**1.75**

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification of Mapping

CO1: Explain operating system concepts including structure, services, process management, and compare architectures of Android, Linux, and Windows.

- **PO1(3):** Students apply fundamental concepts of computer science, operating systems, process management, system calls, kernel architecture, and resource management. The course develops the engineering knowledge required to understand how modern operating systems function and support computing systems.
- **PO2(2):** Students analyze different operating system architectures and identify the strengths and limitations of Android, Linux, and Windows. They examine process behavior, scheduling requirements, and system services to understand system-level problems.
- **PO3(1):** Students gain introductory knowledge about operating system design principles, kernel structures, and system organization, which helps them understand how operating system components are designed to satisfy user and system requirements.
- **PO5(1):** Students use virtualization tools, operating system simulators, Docker, QEMU, and system monitoring utilities to study operating system behavior and architecture.
- **PO10(1):** Students prepare reports, architecture comparisons, and presentations related to different operating systems and communicate technical findings effectively.
- **PO12(1):** Operating systems continuously evolve. Students develop the ability to learn emerging operating systems, virtualization technologies, and cloud-native platforms independently.
- **PSO1(2):** IoT devices rely on operating systems such as Linux, Android Things, FreeRTOS, and embedded operating systems. Understanding OS fundamentals helps students design and optimize IoT platforms.
- **PSO2(1):** Basic understanding of operating system services, process isolation, permissions, and access control mechanisms provides foundational knowledge for secure system design.

CO2: Apply process scheduling and synchronization techniques to solve critical section problems and analyze deadlock handling methods.

- **PO1(3):** Students apply concepts of CPU scheduling algorithms, synchronization primitives, semaphores, monitors, mutexes, and deadlock management using engineering fundamentals.

- **PO2(3):** Students identify and analyze concurrency problems, race conditions, starvation, deadlocks, and resource allocation issues using systematic problem-solving approaches.
- **PO3(2):** Students design synchronization mechanisms and scheduling strategies to improve system performance and ensure correct execution of concurrent processes.
- **PO4(1):** Students perform experiments using different scheduling algorithms and synchronization techniques, analyze system behavior, and compare performance metrics.
- **PO5(2):** Students use simulation tools, Linux process management utilities, and operating system laboratories to evaluate scheduling and synchronization mechanisms.
- **PO9(1):** Many synchronization and concurrency experiments are performed in laboratory groups where students collaborate to solve process management problems.
- **PO12(1):** Students explore advanced concepts such as multicore scheduling, cloud resource allocation, and distributed synchronization, encouraging continuous learning.
- **PSO1(2):** Efficient scheduling and synchronization are essential for real-time IoT systems where multiple sensors and devices operate simultaneously.
- **PSO2(2):** Synchronization and mutual exclusion are important for secure transaction processing, blockchain consensus mechanisms, and secure distributed computing environments.

CO3: Analyze memory management techniques and file system structures, and evaluate system performance using appropriate tools.

- **PO1(2):** Students apply concepts of memory allocation, paging, segmentation, virtual memory, page replacement algorithms, and file system organization.
- **PO2(3):** Students analyze memory utilization, page faults, fragmentation, storage organization, and system performance issues to determine optimal solutions.
- **PO3(2):** Students evaluate and design suitable memory management strategies and file system approaches based on application requirements.
- **PO4(2):** Students conduct experiments on page replacement algorithms, memory utilization, file system performance, and storage management techniques.
- **PO5(3):** Students use memory monitoring tools, performance analyzers, Linux utilities, virtual machines, and benchmarking software to evaluate system performance.
- **PO10(1):** Students document observations, performance results, and comparative analyses through technical reports and presentations.
- **PO12(1):** Advancements in memory technologies, cloud storage, distributed file systems, and virtualization encourage students to engage in continuous learning.
- **PSO1(2):** Memory optimization and efficient storage management are critical for resource-constrained IoT devices and edge computing platforms.
- **PSO2(2):** Secure file systems, memory protection mechanisms, and storage management contribute to secure computing and blockchain infrastructure.

CO4: Explain distributed system concepts and implement communication and coordination mechanisms using technologies like RPC and gRPC.

- **PO1(2):** Students apply concepts of distributed computing, communication protocols, remote procedure calls, and distributed architectures.

- **PO2(2):** Students analyze challenges such as communication delays, synchronization issues, fault tolerance, and distributed resource management.
- **PO3(3):** Students design and implement distributed communication mechanisms using RPC and gRPC to support scalable distributed applications.
- **PO4(2):** Students conduct experiments on distributed communication systems, evaluate performance, and interpret experimental results.
- **PO5(3):** Students use modern distributed computing frameworks, gRPC libraries, networking tools, Docker containers, and cloud-based platforms.
- **PO6(1):** Distributed systems support applications in healthcare, smart cities, e-governance, and industrial automation, requiring awareness of societal needs.
- **PO9(1):** Distributed system projects often require collaborative development and integration among multiple team members.
- **PO10(2):** Students communicate technical designs, protocols, distributed architectures, and implementation details through reports and presentations.
- **PO11(1):** Distributed application development requires planning, resource allocation, and coordination among team members during project execution.
- **PO12(1):** Rapid developments in cloud computing, microservices, edge computing, and distributed architectures encourage continuous professional development.
- **PSO1(3):** Distributed communication, RPC, gRPC, and service-oriented architectures form the backbone of modern IoT, edge computing, and smart device ecosystems.
- **PSO2(2):** Distributed systems concepts directly support blockchain networks, decentralized applications, secure communication protocols, and distributed ledger technologies.

FF No. : 654

MM1501D: Product Lifecycle Management

Credits: 3 **Teaching Scheme:** Theory: 2 hours/Week, Tutorial: 1Hour/Week

Course Objectives:

1. Understand the fundamental concepts, architecture, and business applications of Product Lifecycle Management (PLM) and Teamcenter.
2. Develop proficiency in using Teamcenter Active Workspace for product data management, product structure management, workflows, and collaboration.
3. Learn Teamcenter installation, deployment, integration, and system administration techniques for enterprise environments.
4. Configure and customize Teamcenter using Business Modeler IDE (BMIDE), Access Manager, Workflow Designer, and Classification tools.
5. Apply Teamcenter administration and customization skills to solve real-world product lifecycle management challenges through practical exercises and industry use cases.

Course Relevance:

This course equips students with the knowledge and practical skills required to manage the complete lifecycle of a product using industry-standard Teamcenter PLM. It enables learners to understand product data management, collaboration, workflow automation, system administration, and customization in enterprise environments. The course prepares students for careers in PLM, digital manufacturing, product engineering, and Industry 4.0 by providing hands-on experience with real-world industrial applications and best practices.

SECTION I

Unit 1: Introduction to PLM and Teamcenter Fundamentals (5 Hours)

Introduction to Product Lifecycle Management (PLM), PLM concepts, benefits and business value, Teamcenter architecture, Teamcenter Foundation, Active Workspace interface, Teamcenter functionalities, Product Data Management (PDM), navigation in Active Workspace, search, object management, folders, properties, reports, audit logs, access rights, collaboration, and industry use cases.

Unit 2: Product Data and Structure Management (5 Hours)

Creation and management of Items, Item Revisions, Datasets, Forms, Named References, Business Objects, Access Management, Check-in/Check-out, Project assignment, Product Structure Explorer (PSE), Assembly Management, BOM creation, Revision Rules, Effectivity, Baselines, Variants,

BOM Compare, Product Structure editing, Relation Browser, Impact Analysis, Workflow basics, Classification, Project Management, Change Management, Microsoft Office Integration, Thin Client, Multi-CAD Integration.

Unit 3: Teamcenter Installation, Deployment and Integration (4 Hours)

Teamcenter architecture (2-Tier/4-Tier), database installation, corporate server installation, License Server, BMIDE installation, Active Workspace installation, Microservices, Web Tier deployment, Apache/IIS/JBoss/WebLogic/WebSphere deployment, Teamcenter directories, FSC/FCC, Performance tuning, Backup & Restore, Replica Sites, SOLR indexing, Active Workspace Launcher, Patching, Upgrading, NX Integration, NX Manager Installation, Attribute Mapping and Teamcenter Integration.

SECTION II

Unit 4: Business Modeling and Customization using BMIDE (5 Hours)

BMIDE interface, Business Objects, Classes, Attributes, Properties, UML, Extension Projects, Deployment methods, Data Model Reports, Custom Objects, Property Constants, Global Constants, Forms, Storage Classes, LOVs, ID Generation Rules, Search Filters, Dataset Types, Product Structure Types, Status Types, Naming Rules, Deep Copy Rules, Action Rules, GRM Rules, Condition Engine, Alias IDs, Complex Properties, Business Modeling customization for enterprise requirements.

Unit 5: Teamcenter Administration and Access Management (5 Hours)

Organization setup, Users, Groups, Roles, Volumes, License Management, User administration in Active Workspace, Queries, Reports, Revision Rules, Access Manager, Rule Tree, Project Administration, Project Security, Object-based and Rule-based Access Control, Project utilities, Workflow Process Modeling, Resource Pools, Workflow Designer, Task Templates, Workflow Troubleshooting, Release Management.

Unit 6: Advanced Teamcenter Administration and Enterprise Applications (4 Hours)

Change Management configuration, Change Types, Workflow integration, XML Rendering Stylesheets, Property Display customization, Classification Administration, Classification Hierarchy, ICO, Import/Export utilities, Enterprise customization, Best practices, Performance optimization, Security, Troubleshooting, Industry implementation case studies and real-world PLM deployment scenarios.

List of Practical's:

1. Explore the Teamcenter Active Workspace interface and customize the user workspace.
2. Perform basic and advanced searches using Quick Search, Saved Search, and Filters
3. Create folders, organize objects, and manage object properties in Active Workspace
4. Create Items, Item Revisions, Documents, Datasets, and Forms.
5. Perform Check-in, Check-out, Revision, and Version management.
6. Perform BOM Compare, Baseline creation, Effectivity, and Revision Rule configuration.
7. Create Assemblies, replace components, and manage substitutes and alternates.
8. Create and execute Workflow Processes using predefined templates.
9. Install and configure Teamcenter and Active Workspace components.
10. Create a BMIDE project and deploy a custom Business Object.
11. Create and configure custom Workflow Templates using Workflow Designer.

List of Tutorials:

1. Introduction to Product Lifecycle Management (PLM)
2. Teamcenter Architecture and Deployment Models
3. Product Data Management
4. Product Structure and BOM Management
5. Workflow, Change Management, and Classification
6. Teamcenter Installation and Enterprise Deployment
7. Business Modeling with BMIDE
8. Administration, Security, and Enterprise Customization

Text Books:

1. M. Grieves, Product Lifecycle Management: Driving the Next Generation of Lean Thinking. New York, NY, USA: McGraw-Hill, 2006.
2. A. Saaksvuori and A. Immonen, Product Lifecycle Management, 3rd ed. Berlin, Germany: Springer, 2008.
3. J. Stark, Product Lifecycle Management: 21st Century Paradigm for Product Realisation, 3rd ed. Cham, Switzerland: Springer, 2015.
4. Siemens Digital Industries Software, Teamcenter Documentation Library (Active Workspace, Teamcenter Administration, BMIDE, Workflow Designer, Access Manager and Installation Guides). Plano, TX, USA: Siemens Digital Industries Software, Latest Edition.

Reference Books:

1. J. Stark, Product Lifecycle Management: Volume 1 – 21st Century Paradigm for Product Realisation. London, U.K.: Springer, 2011.
2. J. Stark, Product Lifecycle Management: Volume 2 – The Devil is in the Details. London, U.K.: Springer, 2011.

3. G. Schuh, S. Rudolf, and A. Assmus, Product Lifecycle Management: Approach, Challenges and Future Trends. Berlin, Germany: Springer, 2014.
4. Siemens Digital Industries Software, Teamcenter Business Modeler IDE (BMIDE) Guide. Plano, TX, USA: Siemens Digital Industries Software, Latest Edition.
5. Siemens Digital Industries Software, Teamcenter Active Workspace Configuration and Administration Guide. Plano, TX, USA: Siemens Digital Industries Software, Latest Edition.
6. Siemens Digital Industries Software, Teamcenter Installation, Deployment and System Administration Guide. Plano, TX, USA: Siemens Digital Industries Software, Latest Edition.
7. Siemens Digital Industries Software, Teamcenter Workflow Designer and Access Manager Guide. Plano, TX, USA: Siemens Digital Industries Software, Latest Edition.

Course Outcomes:

After completion of the course, student will be able to:

CO1: Explain PLM concepts, Teamcenter architecture, Active Workspace, and enterprise product data management processes.

CO2: Create, manage, and organize engineering data, product structures, revisions, workflows, projects, and access control using Teamcenter Active Workspace.

CO3: Install, configure, deploy, and troubleshoot Teamcenter server, Active Workspace, indexing, and integration components in enterprise environments.

CO4: Develop and customize Teamcenter data models using BMIDE, including business objects, properties, forms, LOVs, workflows, and access management.

CO5: Administer Teamcenter environments by managing users, roles, projects, workflows, classification, change management, and enterprise customization to meet industrial PLM requirements.

CO-PO Mapping

CO	Program Outcomes (PO)												PSO	
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	-	2	-	-	-	-	1	-	2	1	1
CO2	2	2	3	-	3	-	-	1	2	2	2	2	2	2
CO3	2	3	2	2	3	-	-	1	2	2	2	2	2	2
CO4	3	3	3	2	3	-	-	1	2	2	2	2	2	3
CO5	2	2	3	1	3	1	1	2	3	2	3	2	2	2
Average	2.4	2.4	2.4	1.7	2.8	1.0	1.0	1.3	2.3	1.8	2.3	2.0	1.8	2.0

Scale:
3 – Strongly Mapped
2 – Moderately Mapped
1 – Slightly Mapped
Blank – Not Mapped

Justification of Mapping:

CO1: Explain PLM concepts, Teamcenter architecture, Active Workspace, and enterprise product data management processes.

- **PO1 (3):** Students apply engineering fundamentals to understand PLM concepts, product lifecycle stages, and Teamcenter architecture.
- **PO2 (2):** Students analyze enterprise product data management processes and identify lifecycle challenges.
- **PO3 (1):** Students gain introductory understanding of enterprise product management solutions.
- **PO5 (2):** Students learn modern PLM software (Teamcenter Active Workspace).
- **PO10 (1):** Students communicate PLM concepts through reports and presentations.
- **PO12 (2):** Students develop awareness of continuously evolving PLM technologies.
- **PSO1(1)**: Basic understanding of lifecycle management supporting IoT product development.
- **PSO2 (1):** Basic understanding of secure product information management relevant to secure system development.

CO2: Create, manage, and organize engineering data, product structures, revisions, workflows, projects, and access control using Teamcenter Active Workspace.

- **PO1 (2):** Application of engineering knowledge in enterprise data management.
- **PO2 (2):** Students analyze product structures, revisions, and workflows.
- **PO3 (3):** Design and organize engineering product structures and workflows.
- **PO5(3):** Extensive use of Teamcenter Active Workspace for enterprise product management.
- **PO8 (1):** Proper access control promotes professional responsibility.
- **PO9 (2):** Team collaboration using shared product data.
- **PO10 (2):** Communication through workflow assignments and documentation.
- **PO11 (2):** Management of engineering projects and product lifecycle activities.
- **PO12 (2):** Encourages continuous learning of PLM tools.
- **PSO1 (2):** Product data management applicable to IoT system development.
- **PSO2 (2):** Access control and engineering data security relate to secure systems.

CO3: Install, configure, deploy, and troubleshoot Teamcenter server, Active Workspace, indexing, and integration components in enterprise environments.

- **PO1 (2):** Applies engineering principles in deployment.
- **PO2 (3):** Diagnoses installation and deployment issues.
- **PO3 (2):** Configures enterprise PLM solutions.
- **PO4(2):** Investigates deployment and integration issues through systematic troubleshooting.
- **PO5 (3):** Extensive use of Teamcenter server, indexing, and enterprise tools.
- **PO8 (1):** Responsible deployment practices.
- **PO9 (2):** Team-based enterprise deployment activities.
- **PO10 (2):** Documentation of deployment procedures.
- **PO11 (2):** Enterprise deployment planning and project coordination.
- **PO12 (2):** Continuous learning of enterprise technologies.
- **PSO1 (2):** PLM deployment supports IoT product lifecycle.
- **PSO2 (2):** Secure deployment aligns with secure enterprise systems.

CO4: Develop and customize Teamcenter data models using BMIDE, including business objects, properties, forms, LOVs, workflows, and access management.

- **PO1 (3):** Strong application of engineering concepts during customization.
- **PO2 (3):** Analyze business requirements for customization.
- **PO3 (3):** Design customized data models and workflows.
- **PO4 (2):** Evaluate customized models and workflows.
- **PO5 (3):** Extensive use of BMIDE and Teamcenter customization tools.
- **PO8 (1):** Ethical handling of enterprise data customization.
- **PO9 (2):** Collaborative development activities.
- **PO10 (2):** Technical documentation and communication.
- **PO11 (2):** Supports enterprise customization projects.
- **PO12 (2):** Continuous learning of enterprise PLM customization.
- **PSO1 (2):** Supports customized IoT product lifecycle solutions.
- **PSO2 (3):** Strong alignment with secure enterprise application development and access management.

CO5: Administer Teamcenter environments by managing users, roles, projects, workflows, classification, change management, and enterprise customization to meet industrial PLM requirements.

- **PO1 (2):** Applies engineering knowledge to enterprise administration.
- **PO2 (2):** Analyzes enterprise administration requirements.
- **PO3 (3):** Designs enterprise workflows and management processes.
- **PO4 (1):** Investigates workflow and change management issues.
- **PO5 (3):** Extensive use of enterprise administration tools.
- **PO6 (1):** Administration includes compliance with organizational policies.
- **PO7 (1):** Efficient lifecycle management minimizes resource wastage.
- **PO8 (2):** Strong emphasis on access control and professional ethics.
- **PO9 (3):** Team collaboration through enterprise workflow management.
- **PO10 (2):** Communication among multidisciplinary engineering teams.
- **PO11 (3):** Strong relation to project management, change management, and enterprise administration.
- **PO12 (2):** Enterprise PLM technologies require continuous learning.
- **PSO1 (2):** Enterprise management of IoT product data.
- **PSO2 (2):** Secure administration supports cybersecurity and blockchain-enabled enterprise systems.

FF No.: 654

CB3205: Design Thinking 3

Credits: 1 **Teaching Scheme:** Theory: -hours/Week, Tutorial: 1 Hours/Week

Prerequisites: Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objectives:

- To provide ecosystem for students and faculty for paper publication and patent filing

Section1:
What is research?
Importance of Paper Publication and Patents
Importance of Paper Publication and Patents Structure of Paper
Journal Publication in conference Literature Review Research Paper Writing
Journal Ratings and Evaluation How to rate a Journal?
Intellectual property (IP) Research Ethics Entrepreneurship

Section2:
Structure of The paper Journal List (Top 50 Journals) Selection of the journal
Use of various online journal selection tools Plagiarism checking
Improving contents of the paper Patent drafting
Patent search Filing of patent Writing answers to reviewer questions Modification in manuscript
Checking of publication draft

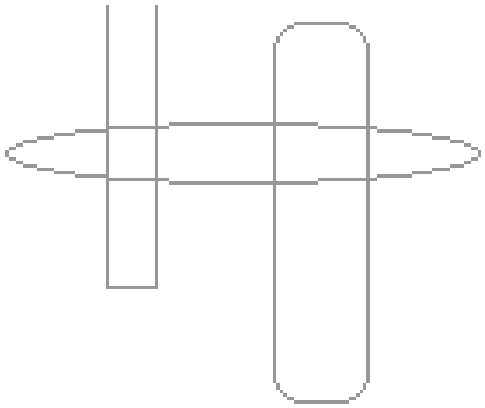
Course Outcomes:
After completion of the course, student will be able to

CO1: Understand the importance of doing Research
CO2: Interpret and distinguish different fundamental terms related to Research
CO3: Apply the methodology of doing research and mode of its publication
CO4: Write a Research Paper based on project work
CO5: Understand Intellectual property rights
CO6: Use the concepts of Ethics in Research , Understand Entrepreneurship and Business Planning

CO – PO / PSO Articulation Matrix

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

CO	Program Outcome (PO)												PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	1	1	1							1	1	2	2
CO2	1	1	1	1	1							1	1	2	2
CO3	2	2	3	3	2	2	1	2	2	3		1	1	3	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	1	3	3
CO5	1	1	1	1	1							1	1	2	2
CO6	2	2	2	2	2	2	1	3	2	3		1	1	3	3
Avg	1.7	1.7	1.8	1.8	1.7	2.0	1.0	2.3	2.0	3.0	1.0	1.0	1.0	2.5	2.5



FF No.: 654

CB3204: Engineering Research & Innovation

Credits: 2 **Teaching Scheme:** Theory: hours/Week, Lab: 2 Hours/Week

Prerequisites:

Course Prerequisites: Research Based Learning Course Relevance: Research Centric Learning (RCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based research projects, curriculum can be enriched with semester long Engineering Research and Innovation courses, in which students can investigate socially relevant problems using various technologies and research methods from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such research problems, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the research domain, thereby achieving research centric learning. Modern world sustained and advanced through the successful completion of research and innovation. In short, if students are prepared for success in life, we need to prepare them for a research-based world. It is a style of active learning and inquiry-based learning. Research centric learning will also redefine the role of teacher as mentor in the learning process. The RCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Course Objectives:

- To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
- To Evaluate alternative approaches, and justify the use of selected tools and methods
- To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
- To engage students in rich and authentic learning experiences.
- To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
- To develop an ecosystem to promote entrepreneurship and research culture among the students

Section 1:

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

Industry Communication Standards, Research Databases and Literature Repositories (IEEE Xplore, Scopus, Google Scholar), Patent Search and IPR Tools, Statistical and Data Analysis Tools, Computational Biology (Biomedical and Bioinformatics), Robotics and Automation, Industry 4.0

(Artificial Intelligence, Machine Learning, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

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

Selection of Research Problem/Topic:
● Students must focus on initiating the task/idea. The idea of inception and consideration shall be from following areas as a real world problem:
-Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
● This is the sample list to start with. Faculty and students are free to include other ar
eas which meet society's requirements at large.
● The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
● A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
● By exemplarity, a problem needs to refer to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

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

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

Developing Inquiry Skills:

● Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What do they want to do with that information?

● They need to analyze information presented within the context of finding answers.

● Modeling is required so that the students can observe and build a conceptual model of the required processes.

● Use the following mechanism to maintain the track of moving towards the solution.

How effective is …….?

How strong is the evidence for ……?

How clear is ……?

● What are the justifications for thinking? Why is the method chosen?

● What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

● It is an integral part of self- directed learning

● Identify the information needed to solve a given problem or issue

● Be able to locate the needed information

● Use the information to solve the given problem effectively.

● Skills required by students in information literacy include:

● · How to prepare for the search? How to carry out research

● Sorting and assessing of information in general

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

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

● Collaborative learning

● Interpersonal Skills

● Resources Evaluation

● Metacognitive Skills

● Reflection Skills

Section 2:

ERI Sample Research Case Studies: -

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

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

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

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

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

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

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

Course Outcomes:

After completion of the course, student will be able to

CO1: Identify a real-life research problem or gap from societal and technological need point of view

CO2: Choose and compare alternative research methodologies to select the most feasible one.

CO3: Analyze and synthesize the identified research problem from a technological and scientific perspective.

CO4: Design a reliable and scalable research methodology/experimental approach to meet the challenges.

CO5: Evaluate the research outcomes based on the criteria specified.

CO6: Inculcate a long life learning and research attitude towards the societal problems.

CO – PO / PSO Articulation Matrix

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

CO	Program Outcome (PO)												PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	2					3		2	2	3	3	2
CO2	2	2	3	2	2		2		3		2	2	3	3	2
CO3	2	2	3	2	3		2		3	1	2	2	3	3	2
CO4	2	2	3	2	3	3		2	3		2	2	3	3	2
CO5	2	2	3	2	3	2			3		2	2	3	3	2
CO6	2	2	3	3	2				3		3	2	3	3	2
Avg	2.0	2.0	2.83	2.83	2.6	2.5	2.0	2.0	3.0	1.0	2.16	2.0	3.0	3.0	2.0