Double Minor Syllabus

Note: The syllabi provided herein for each subject are indicative and intended solely for reference purposes. As these courses are to be delivered through online platforms such as MOOC/NPTEL or other approved content providers, the exact alignment of content with the prescribed syllabus may not always be feasible. Therefore, the contents should be considered as a notional framework only.

The concerned Head of the Department, along with the duly constituted committee responsible for the curation and approval of online courses opted by students, shall exercise due diligence in reviewing and approving the selected courses, ensuring that the learning outcomes are in reasonable conformity with the reference syllabus provided.

Computer Science and Engineering

Track: Cyber Security (Honours / Double Minor)

SEM III	3CS245DH1	Fundamentals of Cyber Security	4	Theory
SEM IV	4CS246DH1	Network & Web Application Security	4	Theory

Syllabus

Subject Code: 3CS245DH1

Subject Title: Fundamentals of Cyber Security

Course Pre-requisite: Basic Computer Fundamentals, Networking Concepts

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of cyber security by being able to do each of the following: • Understand the core concepts of cybersecurity and its importance. • Identify different types of cyber threats and vulnerabilities. • Learn basic cryptographic techniques and security protocols. • Understand access control, authentication, and security mechanisms. • Explore real-world use cases of cybersecurity in systems and applications.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Define and explain key cybersecurity principles and terminology.
2. Identify common cyber threats, attacks, and vulnerabilities.
3. Describe various cryptographic algorithms and their applications.
4. Understand security policies, controls, and user authentication.
5. Recognize secure software and system design practices.
6. Apply cybersecurity principles to protect information systems.

Syllabus:

Unit I: Introduction to Cyber Security (Hours: 07) Definition and goals, security principles (CIA), threat landscape, cybersecurity domains.

Unit II: Cyber Threats and Vulnerabilities (Hours: 07) Malware types, phishing, ransomware, social engineering, vulnerability types and lifecycle.

Unit III: Cryptography Basics (Hours: 07) Symmetric and asymmetric encryption, hashing, digital signatures, key exchange methods.

Unit IV: Network Security (Hours: 07) Firewalls, IDS/IPS, VPN, common network attacks and mitigation strategies.

Unit V: Authentication and Access Control (Hours: 07) User authentication techniques (password, biometric, multifactor), access control models (DAC, MAC, RBAC).

Unit VI: Cyber Security in Practice (Hours: 07) Secure software development, incident response basics, cyber law overview, introduction to ethical hacking.

Text Books:

1. William Stallings, “Network Security Essentials,” Pearson.
2. Nina Godbole and Sunit Belapure, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives,” Wiley.

Reference Books:

1. Michael T. Goodrich, Roberto Tamassia, “Introduction to Computer Security,” Pearson.
2. Chuck Easttom, “Computer Security Fundamentals,” Pearson.
3. Eric Maiwald, “Fundamentals of Network Security,” McGraw-Hill.
4. Behrouz A. Forouzan, “Cryptography and Network Security,” McGraw-Hill.
5. Official NIST Cybersecurity Framework Documentation

Subject Code: 4CS246DH1

Subject Title: Network & Web Application Security

Course Pre-requisite: Fundamentals of Cyber Security, Basic Networking Concepts

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of network and web application security by being able to do each of the following: • Learn about the threats and countermeasures in computer networks. • Understand the protocols and tools used in securing networks. • Explore web application vulnerabilities and mitigation techniques. • Understand secure web development practices. • Implement security testing tools and frameworks.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Understand the nature of network-based threats and attacks.
2. Apply security mechanisms for securing network infrastructure.
3. Identify web application security issues.
4. Implement secure coding practices in web applications.

5. Use tools for web and network security testing.
6. Assess and mitigate risks in network and web environments.

Syllabus:

Unit I: Network Security Fundamentals (Hours: 07) Network vulnerabilities and threats, OSI & TCP/IP security layers, network security controls.

Unit II: Protocol Security (Hours: 07) IPSec, SSL/TLS, HTTPS, DNSSEC, Email security protocols (SPF, DKIM).

Unit III: Firewalls and Intrusion Detection Systems (Hours: 07) Types of firewalls, IDS/IPS architecture, anomaly vs signature-based detection, SIEM.

Unit IV: Web Application Threats and OWASP Top 10 (Hours: 07) Injection, XSS, CSRF, insecure deserialization, broken authentication and access control.

Unit V: Secure Web Development (Hours: 07) Input validation, session management, secure cookies, error handling, secure APIs.

Unit VI: Web Security Testing Tools and Frameworks (Hours: 07) Burp Suite, OWASP ZAP, Nikto, SQLMap, static and dynamic analysis tools.

Text Books:

1. William Stallings, "Network Security Essentials: Applications and Standards," Pearson.
2. Dafydd Stuttard & Marcus Pinto, "The Web Application Hacker's Handbook," Wiley.

Reference Books:

1. Neil Rowe, "Introduction to Cybersecurity: Networking and Security," Springer.
2. Joseph Migga Kizza, "Computer Network Security," Springer.
3. OWASP Testing Guide and OWASP Top 10 Documentation.
4. Mike Shema, "Web Application Security," Syngress.
5. Nina Godbole, "Information Systems Security," Wiley.

Track: Data Science (Honours / Double Minor)

SEM III	3CS245DH2	Introduction to Data Science	4	Theory
SEM IV	4CS246DH2	Statistical Methods for Data Analysis	4	Theory

Syllabus

Subject Code: 3CS245DH2

Subject Title: Introduction to Data Science

Course Pre-requisite: Computer Programming, Mathematics for Computing

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of data science concepts by being able to do each of the following:
• Understand the data science process and its applications.
• Learn data preprocessing and data visualization.
• Understand data types and sources.
• Explore real-world datasets and extract insights.
• Use Python for data analysis.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:

1. Understand the role and scope of data science.
2. Perform data preprocessing and exploratory analysis.
3. Work with structured and unstructured data.
4. Apply visualization techniques for insight generation.
5. Use Python libraries for data science (Pandas, NumPy, Matplotlib).
6. Understand real-life applications of data science.

Syllabus:

Unit I: Data Science Overview (Hours: 07)
Definition, lifecycle, applications, roles in data science.

Unit II: Data Types and Sources (Hours: 07)
Structured vs unstructured data, Datasets, APIs, Web scraping basics.

Unit III: Data Preprocessing (Hours: 07)
Data cleaning, transformation, handling missing data, normalization.

Unit IV: Exploratory Data Analysis (Hours: 07)
Descriptive statistics, visualization techniques, outlier detection.

Unit V: Introduction to Python for Data Science (Hours: 07)
Pandas, NumPy, Matplotlib, DataFrames, basic plots.

Unit VI: Mini Project and Case Studies (Hours: 07)
Mini project on real-world data, case studies from health, finance, e-commerce.

Text Books:

1. Cathy O'Neil and Rachel Schutt, “Doing Data Science”, O’Reilly Media.
2. Joel Grus, “Data Science from Scratch”, O’Reilly.

Reference Books:

1. Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, O'Reilly.
2. Vasant Dhar, “Data Science and Prediction”, Communications of the ACM.

3. Hadley Wickham, “R for Data Science”, O’Reilly.
4. Wes McKinney, “Python for Data Analysis”, O’Reilly.
5. IBM Data Science Course (MOOC).

Subject Code: 4CS246DH2

Subject Title: Statistical Methods for Data Analysis

Course Pre-requisite: Introduction to Data Science, Mathematics for Computing

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of statistical techniques in data analysis by being able to do each of the following:
• Understand fundamental concepts of statistics.
• Apply descriptive and inferential statistical techniques.
• Perform hypothesis testing and correlation analysis.
• Use regression models for prediction.
• Analyze and interpret statistical results using tools.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:

1. Understand and apply basic descriptive statistics.
2. Perform probability analysis and hypothesis testing.
3. Apply regression and correlation analysis techniques.
4. Evaluate statistical significance and draw conclusions.
5. Use Python/R for performing statistical computations.
6. Interpret results and communicate statistical findings.

Syllabus:

Unit I: Descriptive Statistics (Hours: 07)
Mean, median, mode, variance, standard deviation, graphical methods.

Unit II: Probability Theory (Hours: 07)
Random variables, distributions, Bayes theorem, discrete and continuous distributions.

Unit III: Statistical Inference (Hours: 07)
Sampling, Central Limit Theorem, Estimation, Confidence Intervals.

Unit IV: Hypothesis Testing (Hours: 07)
T-tests, Chi-square tests, ANOVA.

Unit V: Correlation and Regression (Hours: 07)
Pearson and Spearman correlation, Linear and logistic regression models.

Unit VI: Statistical Tools & Applications (Hours: 07)
Hands-on with Python/R, interpretation, case studies from scientific and business domains.

Text Books:

1. S. C. Gupta & V. K. Kapoor, "Fundamentals of Mathematical Statistics", Sultan Chand.
2. Allen B. Downey, "Think Stats: Probability and Statistics for Programmers", O'Reilly.

Reference Books:

1. Larry J. Stephens, "Essential Statistics", Cengage.
2. Andy Field, "Discovering Statistics Using R", Sage.
3. Peter Bruce & Andrew Bruce, "Practical Statistics for Data Scientists", O'Reilly.
4. John Verzani, "Using R for Introductory Statistics", CRC Press.
5. NPTEL Course Materials on Statistics for Data Science.

Computer Science and Engineering (Data Science)

Track: Artificial Intelligence (AI) (Honours / Double Minor)

SEM III	3DS245DH1	Fundamentals of Artificial Intelligence	4	Theory
SEM IV	4DS246DH1	Intelligent Systems and Applications	4	Theory

Syllabus

Subject Code: 3DS245DH1

Subject Title: Fundamentals of Artificial Intelligence
Credits: 4
Type: Theory

Course Pre-requisite:
- Basic Programming Concepts
- Discrete Mathematics and Logic
- Problem Solving Techniques

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of foundational concepts in Artificial Intelligence (AI) by being able to do each of the following:
• Understand the fundamental principles of AI and its historical evolution.
• Explore core techniques such as search strategies, logic-based reasoning, and knowledge representation.
• Introduce basic machine learning concepts.
• Understand applications of AI in various domains.
• Learn the ethical and societal impacts of AI technologies.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:

1. Define AI, its goals, and applications across sectors.
2. Solve problems using uninformed and informed search techniques.
3. Apply logical reasoning using propositional and predicate logic.
4. Understand basic concepts of machine learning and expert systems.
5. Represent knowledge using various representation techniques.
6. Discuss ethical considerations and future trends in AI.

Syllabus:
Unit I: Introduction to Artificial Intelligence (Hours: 07)
History and evolution of AI, Definitions and scope, Turing Test, Applications of AI (Robotics, NLP, Healthcare, Games, etc.), Intelligent agents and environments.
Unit II: Problem Solving and Search Techniques (Hours: 07)
Problem formulation, State space, Uninformed search (BFS, DFS, UCS), Informed search (Greedy, A*), Constraint satisfaction problems.
Unit III: Knowledge Representation and Reasoning (Hours: 07)
Propositional logic, Predicate logic, Resolution, Forward and backward chaining, Rule-based systems, Semantic networks and frames.
Unit IV: Introduction to Machine Learning (Hours: 07)
Supervised vs Unsupervised learning, Regression, Classification basics, Clustering overview, Decision Trees, Naive Bayes, k-NN.
Unit V: Expert Systems and Planning (Hours: 07)
Architecture of expert systems, Inference engines, Case-based reasoning, Classical planning, Planning graphs, STRIPS.
Unit VI: AI Ethics and Societal Implications (Hours: 07)
Ethical issues in AI, Bias in data and algorithms, Explainable AI, AI safety, Job displacement, Future of AI.

Text Books:

1. Elaine Rich, Kevin Knight, Shivashankar B. Nair, “Artificial Intelligence”, McGraw Hill.
2. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson.

Reference Books:

1. Nils J. Nilsson, “Principles of Artificial Intelligence”, Morgan Kaufmann.
2. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press.
3. Dan W. Patterson, “Introduction to AI and Expert Systems”, Pearson.
4. Peter Flach, “Machine Learning: The Art and Science of Algorithms”, Cambridge.
5. IBM AI Foundations Courseware (MOOC).

Subject Code: 4DS246DH2
Subject Title: Intelligent Systems and Applications

Credits: 4
Type: Theory

Course Pre-requisite:

- Fundamentals of Artificial Intelligence
- Programming Knowledge in Python
- Basic Concepts of Logic and Algorithms

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of intelligent systems by being able to do each of the following:

• Understand the architecture and functioning of intelligent systems.
• Explore components such as sensors, actuators, agents, and environments.
• Learn design and development approaches for intelligent systems.
• Study real-life intelligent systems applications.
• Familiarize with AI-driven decision-making processes and adaptive systems.

Course Outcomes (Expected Outcome):

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

1. Define intelligent systems and describe their core components.
2. Design agent-based models for intelligent decision-making.
3. Implement logic-based and rule-based intelligent systems.
4. Analyze adaptive systems using reinforcement and fuzzy logic.
5. Apply intelligent system techniques in robotics, IoT, and automation.
6. Evaluate real-world intelligent applications in various domains.

Syllabus:

Unit I: Introduction to Intelligent Systems (Hours: 07)

Definition and characteristics, Comparison with traditional systems, Types of intelligent systems, Intelligent agents and environments.

Unit II: Agent-Based Systems (Hours: 07)

Agent types (reflex, goal-based, utility-based), Multi-agent systems, Architecture of agent-based systems, Communication among agents.

Unit III: Rule-Based and Expert Systems (Hours: 07)

Rule representation, Inference mechanisms, Production systems, Knowledge-based expert systems, Forward and backward chaining.

Unit IV: Fuzzy Logic and Adaptive Systems (Hours: 07)

Fuzzy sets and operations, Fuzzy inference system, Applications of fuzzy logic, Basics of reinforcement learning, Feedback and learning loops.

Unit V: Intelligent System Applications in Robotics & IoT (Hours: 07)

Robotic perception and planning, Embedded intelligence, Sensor integration, Edge intelligence in IoT environments, Smart automation.

Unit VI: Case Studies and Emerging Applications (Hours: 07)

Intelligent healthcare systems, Smart transportation, E-commerce recommendation engines, Industrial automation, Smart cities.

Text Books:

1. David Poole and Alan Mackworth, “Artificial Intelligence: Foundations of Computational Agents”, Cambridge University Press.
2. Rajendra Akerkar, “Intelligence Systems”, PHI Learning.

Reference Books:

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Pearson.
2. Elaine Rich and Kevin Knight, “Artificial Intelligence”, McGraw Hill.

3. Timothy J. Ross, “Fuzzy Logic with Engineering Applications”, Wiley.
4. Nils J. Nilsson, “Artificial Intelligence: A New Synthesis”, Morgan Kaufmann.
5. IEEE/ACM Case Studies on Intelligent Systems (Open Access Resources).

Track: Data Science (Honours / Double Minor)

SEM III	3DS245DH2	Data Science Foundations and Tools	4	Theory
SEM IV	4DS246DH2	Statistical Data Modelling and Inference	4	Theory

Syllabus

Subject Code: 3DS245DH2
Subject Title: Data Science Foundations and Tools
Credits: 4
Type: Theory

Course Pre-requisite:
- Computer Programming
- Basic Mathematics for Data Analysis

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of foundational concepts and tools in data science by being able to do each of the following:
• Understand the fundamental concepts and lifecycle of data science.
• Learn data wrangling, preprocessing, and cleaning techniques.
• Develop foundational Python skills for data analysis.
• Explore common data science libraries and tools.
• Work on real-world datasets using Jupyter Notebooks.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Define the scope, lifecycle, and goals of data science.
2. Perform basic data collection, exploration, and cleaning.
3. Use Python and its libraries for data manipulation and analysis.
4. Apply data visualization techniques using tools like Matplotlib and Seaborn.
5. Understand the role of data science in decision-making.
6. Build simple data-driven applications using open-source toolkits.

Syllabus:
Unit I: Introduction to Data Science (Hours: 07)
What is Data Science? Data science lifecycle, Roles (Data Analyst, Data Engineer, Data Scientist), Applications in different domains, Skills required for a data scientist.
Unit II: Python for Data Science (Hours: 07)
Python basics – variables, loops, conditionals, functions; Python libraries – NumPy, Pandas; DataFrames and Series; Reading and writing CSV/Excel/JSON files.
Unit III: Data Wrangling and Preprocessing (Hours: 07)
Handling missing data, Outliers, Encoding categorical variables, Feature scaling and transformation, String operations, Date-time handling.
Unit IV: Data Visualization (Hours: 07)

Introduction to Matplotlib and Seaborn, Histograms, Box plots, Pair plots, Heatmaps, Correlation matrices, Plot styling and labeling.

Unit V: Introduction to Databases and SQL (Hours: 07)

Relational database concepts, SQL queries (SELECT, JOIN, GROUP BY), Connecting Python with SQLite and MySQL using libraries.

Unit VI: Case Studies and Mini Project (Hours: 07)

Mini-project based on real-world dataset: health, social media, e-commerce, finance; Case studies demonstrating end-to-end data science workflows.

Text Books:

1. Joel Grus, “Data Science from Scratch”, O'Reilly Media.
2. Wes McKinney, “Python for Data Analysis”, O’Reilly Media.

Reference Books:

1. Cathy O’Neil and Rachel Schutt, “Doing Data Science”, O’Reilly Media.
2. Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow”, O’Reilly.
3. Hadley Wickham, “R for Data Science”, O’Reilly.
4. IBM Data Science Handbook (MOOC Resources).
5. Microsoft Learn and Kaggle Datasets for practice.

Subject Code: 4DS246DH2

Subject Title: Statistical Data Modelling and Inference

Credits: 4

Type: Theory

Course Pre-requisite:

- Data Science Foundations and Tools
- Applied Mathematics
- Probability and Basic Statistics

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of statistical modeling and inference by being able to do each of the following:

- Understand the role of statistics in data-driven problem solving.
- Learn probability theory, distributions, and hypothesis testing.
- Develop skills for statistical inference and regression modeling.
- Apply linear and logistic regression for real-world data analysis.
- Interpret and communicate statistical findings effectively.

Course Outcomes (Expected Outcome):

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

1. Understand core concepts of probability and statistical reasoning.
2. Apply descriptive and inferential statistics to analyze data.
3. Formulate and test hypotheses using appropriate statistical methods.
4. Build and evaluate linear and logistic regression models.
5. Work with real-world data using statistical software tools.
6. Interpret statistical results and communicate insights effectively.

Syllabus:

Unit I: Probability and Random Variables (Hours: 07)
Basic probability theory, Conditional probability, Bayes' theorem, Random variables (discrete & continuous), Expectation and variance.
Unit II: Probability Distributions (Hours: 07)
Common distributions: Binomial, Poisson, Normal, Exponential; Central Limit Theorem; Applications in modeling real-world events.
Unit III: Statistical Inference (Hours: 07)
Sampling, Point and interval estimation, Confidence intervals, Properties of estimators, Law of large numbers.

Unit IV: Hypothesis Testing (Hours: 07)
Null and alternative hypotheses, Type I and II errors, p-values, z-test, t-test, Chi-square test, ANOVA; Application scenarios.
Unit V: Regression Modelling (Hours: 07)
Simple and multiple linear regression, Model assumptions, Goodness-of-fit (R^2, Adjusted R^2), Multicollinearity, Residual analysis.
Unit VI: Logistic Regression and Applications (Hours: 07)
Binary classification, Odds and log-odds, Maximum likelihood estimation, ROC curve and AUC, Real-life applications and case studies.

Text Books:

1. Douglas C. Montgomery & George C. Runger, "Applied Statistics and Probability for Engineers", Wiley.
2. Larry Wasserman, "All of Statistics: A Concise Course in Statistical Inference", Springer.

Reference Books:

1. Allen B. Downey, "Think Stats: Probability and Statistics for Programmers", O'Reilly.
2. Freedman, Pisani, Purves, "Statistics", Norton.
3. Joseph Schmuller, "Statistical Analysis with Excel for Dummies", Wiley.
4. William Mendenhall et al., "Introduction to Probability and Statistics", Cengage.
5. JASP and R (Software Tools for Statistical Analysis – Open Source)

Artificial Intelligence & Data Science

Track: Blockchain Technologies (Honours / Double Minor)

SEM III	3AD245DH1	Foundations of Blockchain	4	Theory
SEM IV	4AD246DH1	Smart Contracts and DApps	4	Theory

Syllabus

Subject Code: 3AD245DH1
Subject Title: Foundations of Blockchain
Credits: 4
Type: Theory

Course Pre-requisite:

- Basics of Computer Networks

- Cryptography fundamentals
- Programming logic and data structures

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of blockchain fundamentals by being able to do each of the following:
• Understand the principles and architecture of blockchain technology.
• Explore key cryptographic and consensus mechanisms that support blockchain.
• Examine the structure and functionality of distributed ledgers.
• Analyze the role of blockchain in ensuring transparency, immutability, and security.
• Study real-world use cases and limitations of blockchain systems.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Describe the foundational components of blockchain architecture.
2. Explain the significance of cryptographic tools in securing blockchain transactions.
3. Differentiate between public, private, and permissioned blockchains.
4. Evaluate consensus mechanisms such as PoW, PoS, and alternatives.
5. Identify use cases and application areas of blockchain across industries.
6. Critically analyze the challenges and future potential of blockchain technologies.

Syllabus:
Unit I: Introduction to Blockchain (Hours: 07)
Definition, origin, and history of blockchain; Evolution of decentralized systems; Characteristics – transparency, decentralization, immutability; Blockchain vs traditional databases.
Unit II: Cryptographic Foundations (Hours: 07)
Public key cryptography, Hash functions (SHA-256), Digital signatures, Merkle trees; Importance of cryptography in blockchain security.
Unit III: Distributed Ledgers and Blockchain Architecture (Hours: 07)
Ledger structures, Blocks and chains, P2P network architecture, Nodes and mining, Blockchain forks, Block propagation and validation.
Unit IV: Consensus Mechanisms (Hours: 07)
Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS, Practical Byzantine Fault Tolerance (PBFT), and emerging consensus models.
Unit V: Blockchain Types and Platforms (Hours: 07)
Public vs Private vs Consortium blockchain; Overview of Ethereum, Hyperledger Fabric, Corda, and Solana.
Unit VI: Blockchain Use Cases and Challenges (Hours: 07)
Applications in finance, supply chain, healthcare, identity management; Scalability, security, energy consumption, legal and ethical issues; Future directions in blockchain.

Text Books:

1. Arvind Narayanan et al., “Bitcoin and Cryptocurrency Technologies”, Princeton University Press.
2. Melanie Swan, “Blockchain: Blueprint for a New Economy”, O'Reilly Media.

Reference Books:

1. Imran Bashir, “Mastering Blockchain”, Packt Publishing.
2. Josh Thompsons, “Blockchain Basics: A Non-Technical Introduction”, CreateSpace.
3. Andreas M. Antonopoulos, “Mastering Bitcoin”, O’Reilly Media.
4. Don Tapscott & Alex Tapscott, “Blockchain Revolution”, Portfolio Penguin.
5. Hyperledger Fabric & Ethereum Official Documentation.

Subject Code: 4AD246DH1
Subject Title: Smart Contracts and DApps
Credits: 4
Type: Theory

Course Pre-requisite:

- Foundations of Blockchain
- Basic knowledge of programming (preferably Python or JavaScript)
- Introduction to Ethereum and decentralized architecture

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of smart contracts and decentralized applications (DApps) by being able to do each of the following:

- Understand the principles and lifecycle of smart contracts.
- Develop smart contracts using Solidity on the Ethereum blockchain.
- Design and deploy decentralized applications using blockchain platforms.
- Explore the interaction between front-end applications and blockchain back-end.
- Evaluate DApps from security, scalability, and usability perspectives.

Course Outcomes (Expected Outcome):

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

1. Understand the architecture and lifecycle of smart contracts.
2. Develop, test, and deploy smart contracts using Solidity and related tools.
3. Create functional DApps using front-end and blockchain integrations.
4. Utilize Ethereum-based tools like Truffle, Ganache, and MetaMask.
5. Analyze the potential vulnerabilities and challenges in smart contract design.
6. Apply blockchain-based applications to real-world use cases.

Syllabus:

Unit I: Introduction to Smart Contracts (Hours: 07)

What are smart contracts? Key characteristics and benefits; Use cases; Limitations and legal implications; Smart contract platforms.

Unit II: Ethereum Ecosystem and Solidity Basics (Hours: 07)

Ethereum Virtual Machine (EVM), Gas mechanism; Introduction to Solidity syntax, data types, functions, contracts, and events.

Unit III: Smart Contract Development and Testing (Hours: 07)

Structs, modifiers, inheritance, libraries, mappings; Contract deployment using Remix IDE; Unit testing with Mocha/Chai and Truffle framework.

Unit IV: Decentralized Applications (DApps) (Hours: 07)

Architecture of DApps – front-end, smart contract, and blockchain layers; Web3.js and Ethers.js; Wallets and interactions using MetaMask.

Unit V: Advanced Concepts and Use Cases (Hours: 07)

Oracles and Chainlink, Inter-contract communication, Upgradable contracts, DAO (Decentralized Autonomous Organization), Token standards (ERC-20, ERC-721).

Unit VI: Security and Best Practices (Hours: 07)

Common vulnerabilities (reentrancy, overflows, denial of service); Security tools and audits; Best practices for smart contract development and deployment.

Text Books:

1. Andreas M. Antonopoulos and Gavin Wood, "Mastering Ethereum", O'Reilly Media.

2. Narayan Prusty, “Building Blockchain Projects”, Packt Publishing.

Reference Books:

1. Ritesh Modi, “Solidity Programming Essentials”, Packt.
2. Kevin Solorio, Randall Kanna, David Hoover, “Hands-On Smart Contract Development with Solidity and Ethereum”, O'Reilly.
3. Ethereum and Solidity Official Documentation.
4. Truffle Suite Documentation.
5. OpenZeppelin Smart Contract Security Guidelines.

Track: Generative AI (Honours / Double Minor)

SEM III	3AD245DH2	Introduction to Generative AI	4	Theory
SEM IV	4AD246DH2	Foundation Models and Transfer Learning	4	Theory

Syllabus

Subject Title: Introduction to Generative AI
Subject Code: 3AD245DH2
Semester: III

Credits: 4
Type: Theory

Course Pre-requisite:

- Fundamentals of Artificial Intelligence
- Python Programming
- Basic Linear Algebra and Probability

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of generative models by being able to do each of the following:

- Understand the fundamentals and types of generative models.
- Explore mathematical foundations behind generative AI techniques.
- Study architectures such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs).
- Analyze the applications of generative AI in creative industries, healthcare, and NLP.
- Introduce tools and platforms commonly used in building generative models.

Course Outcomes (Expected Outcome):

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

1. Define and differentiate between discriminative and generative models.
2. Explain the architecture and training process of VAEs and GANs.
3. Apply generative models for image, audio, and text generation.
4. Understand loss functions and optimization techniques for generative models.
5. Describe real-world use cases of generative AI in various sectors.
6. Get hands-on experience with libraries like TensorFlow, PyTorch, and HuggingFace.

Syllabus:

Unit I: Fundamentals of Generative Models (Hours: 07)
Introduction to Generative AI; Discriminative vs. Generative Models; Overview of unsupervised learning; Introduction to latent space representation.
Unit II: Mathematical Foundations (Hours: 07)
Probability theory, KL divergence, Bayesian inference, Maximum likelihood estimation; Optimization methods: SGD, Adam.
Unit III: Variational Autoencoders (VAEs) (Hours: 07)
Architecture of VAEs; Encoder-Decoder Framework; Reparameterization trick; Loss functions; Applications in image reconstruction.
Unit IV: Generative Adversarial Networks (GANs) (Hours: 07)
GAN architecture; Generator and Discriminator; Training dynamics; Common issues (mode collapse, convergence); DCGAN and conditional GANs.
Unit V: Applications of Generative AI (Hours: 07)
Image synthesis, Text generation, Music & art generation, DeepFakes, Data augmentation, Medical imaging applications.
Unit VI: Tools, Libraries and Trends (Hours: 07)
TensorFlow, PyTorch, HuggingFace, OpenAI APIs; Google Colab for experimentation; Ethical considerations and safety in generative AI.

Text Books:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press.
2. David Foster, *Generative Deep Learning*, O'Reilly Media.

Reference Books:

1. Francois Chollet, *Deep Learning with Python*, Manning.
2. Sebastian Raschka, *Machine Learning with PyTorch and Scikit-Learn*, Packt.
3. Carl Doersch, *Tutorial on Variational Autoencoders*.
4. Papers from arXiv.org on GANs and VAEs.
5. HuggingFace and TensorFlow official documentation.

Subject Title: Foundation Models and Transfer Learning
Subject Code: 4AD246DH2
Semester: IV
Credits: 4
Type: Theory

Course Pre-requisite:

- Introduction to Generative AI
- Deep Learning Fundamentals
- Knowledge of Python and ML libraries (PyTorch/TensorFlow)

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of foundation models and transfer learning by being able to:
• Understand the concept and architecture of foundation models.
• Explore techniques in transfer learning, fine-tuning, and domain adaptation.
• Study the impact of large-scale pretraining in NLP and vision.
• Analyze transformer-based architectures and their applications.
• Work with APIs and platforms offering pre-trained models.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Explain what foundation models are and their role in modern AI systems.
2. Understand transformer architecture and pretraining techniques.
3. Apply transfer learning approaches like feature extraction and fine-tuning.
4. Use pretrained models for NLP, vision, and multimodal tasks.
5. Identify the trade-offs and challenges in deploying foundation models.
6. Experiment with models like BERT, GPT, and CLIP using open-source tools.

Syllabus:
Unit I: Introduction to Foundation Models (Hours: 07)
Definition, characteristics, and emergence; Pretraining and generalization; Examples: BERT, GPT, DALL·E, T5.
Unit II: Transformer Architecture (Hours: 07)
Attention mechanisms, self-attention, multi-head attention; Position encoding; Encoder-decoder models.
Unit III: Transfer Learning Techniques (Hours: 07)
Feature-based vs. fine-tuning approaches; Domain adaptation; Few-shot and zero-shot learning.
Unit IV: NLP Applications using Foundation Models (Hours: 07)
Text summarization, question answering, translation, sentiment analysis; HuggingFace Transformers; GPT-2/3 use cases.
Unit V: Vision and Multimodal Models (Hours: 07)
Vision Transformers (ViT), CLIP, DALL·E, BLIP; Pretrained models for image classification and captioning.
Unit VI: Tools, APIs and Deployment (Hours: 07)
HuggingFace, OpenAI, Google's T5 and PaLM APIs; Model evaluation, prompt engineering basics; Model cards and ethical considerations.

Text Books:
1. Thomas Wolf et al., *Transformers for Natural Language Processing*, O'Reilly.
2. Kevin Murphy, *Probabilistic Machine Learning: Advanced Topics*, MIT Press.

Reference Books:
1. Vaswani et al., *Attention is All You Need*, NeurIPS 2017.
2. Sebastian Ruder, *Transfer Learning Tutorial Series*.
3. Papers with Code: https://paperswithcode.com
4. HuggingFace Documentation and Model Hub
5. OpenAI API Docs and Research Papers

Computer Engineering

Track: Internet of Things (IoT) (Honours / Double Minor)

SEM III	3KE245DH1	Fundamentals of IoT and Smart Devices	4	Theory
SEM IV	4KE246DH1	IoT Architectures and Communication Protocols	4	Theory

Syllabus

Subject Code: 3KE245DH1

Subject Title: Fundamentals of IoT and Smart Devices
Semester: III
Credits: 4
Type: Theory

Course Pre-requisite:
Basic knowledge of electronics, computer fundamentals, and programming (C/Python preferred).

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of IoT and smart device fundamentals by being able to:
• Understand the scope, applications, and components of the Internet of Things (IoT).
• Study the architecture of IoT systems including sensors, actuators, and microcontrollers.
• Learn basic interfacing of hardware with software platforms.
• Develop foundational skills in device communication and control.
• Explore examples of smart devices in real-world IoT ecosystems.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Describe the structure and applications of IoT in different domains.
2. Identify the basic components of IoT systems (hardware and software).
3. Understand the functioning of sensors, actuators, and microcontrollers.
4. Interface simple sensors and output devices with a controller.
5. Build and simulate simple IoT projects using open-source platforms.
6. Analyze data collection and basic decision logic in smart devices.

Syllabus:
Unit I: Introduction to IoT (Hours: 07)
Definition, history and evolution, characteristics, trends, and applications of IoT in smart homes, agriculture, healthcare, and industry.
Unit II: IoT Building Blocks (Hours: 07)
Core elements: Things, Gateways, Network, Cloud, Data Analytics; Sensors and actuators, Edge devices, Microcontrollers (Arduino, NodeMCU).
Unit III: Microcontrollers and Embedded Systems (Hours: 07)
Architecture of microcontrollers, Introduction to Arduino/Raspberry Pi/ESP32, GPIO programming, working with development boards.
Unit IV: Sensor Interfacing and Data Acquisition (Hours: 07)
Types of sensors (temperature, light, proximity, humidity), analog/digital signal basics, ADC, sensor calibration, I/O programming.
Unit V: Actuators and Device Control (Hours: 07)
DC motors, servo motors, buzzers, relays, and display devices. Basic interfacing for real-time output.
Unit VI: Smart Devices and Project Demonstrations (Hours: 07)
Designing simple smart devices (e.g., smart light, smart fan), logic implementation, data flow from sensor to actuator, simulation using platforms like Tinkercad or Proteus.

Text Books:

1. Arshdeep Bahga, Vijay Madisetti, *Internet of Things: A Hands-on Approach*, Universities Press.
2. Raj Kamal, *Internet of Things: Architecture and Design*, McGraw Hill.

Reference Books:

1. Jan Holler et al., *From Machine-to-Machine to the Internet of Things*, Elsevier.

2. Pethuru Raj & Anupama C. Raman, *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, CRC Press.
3. Marco Schwartz, *Internet of Things with the Arduino Yun*, Packt.
4. Michael Margolis, *Arduino Cookbook*, O'Reilly.
5. Tutorials and datasheets from Arduino.cc and Espressif.com.

Subject Code: 4KE246DH1
Subject Title: IoT Architectures and Communication Protocols
Semester: IV
Credits: 4
Type: Theory
Course Pre-requisite:
Fundamentals of IoT and Smart Devices, Basic Networking Concepts.

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of IoT system architectures and communication protocols by being able to:
• Understand layered architectures and functional models in IoT.
• Explore key wireless and wired communication protocols used in IoT.
• Study device-to-device and device-to-cloud communication patterns.
• Analyze protocol stacks such as MQTT, CoAP, HTTP, LoRaWAN.
• Investigate standards, interoperability, and protocol selection for use-case-driven IoT systems.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Explain the different layers and models used in IoT architectures.
2. Differentiate between various IoT communication protocols.
3. Design basic communication flows for device-to-device/cloud interaction.
4. Use lightweight protocols (MQTT/CoAP) in real-time data applications.
5. Understand the challenges in protocol selection for scalable IoT systems.
6. Apply interoperable protocol stacks in constrained networks.

Syllabus:
Unit I: IoT System Architectures (Hours: 07)
IoT reference architecture (three-layer, five-layer, and service-oriented), Functional view, Application view, Deployment view, Fog and edge layers.
Unit II: Networking Basics for IoT (Hours: 07)
OSI & TCP/IP models, MAC addressing, IPv4 vs IPv6, Routing in IoT, Network topologies and QoS issues.
Unit III: Wireless Communication Protocols (Hours: 07)
Wi-Fi, Bluetooth/BLE, Zigbee, LoRa, NB-IoT, RFID/NFC – working, pros/cons, and use-case suitability.
Unit IV: Device Communication Protocols (Hours: 07)
Serial communication (UART, I2C, SPI), Device-to-device, Sensor-actuator messaging, data bus communication.
Unit V: Application Layer Protocols (Hours: 07)
HTTP, MQTT, CoAP – features, comparison, RESTful APIs, protocol stacks for low-power networks.
Unit VI: Security and Interoperability in Protocols (Hours: 07)
Protocol vulnerabilities, Transport Layer Security (TLS), Interoperability challenges in heterogeneous IoT, protocol standardization efforts (IETF, IEEE, W3C).

Text Books:

1. Olivier Hersent, David Boswarthick, Omar Elloumi, *The Internet of Things: Key Applications and Protocols*, Wiley.
2. Arshdeep Bahga, Vijay Madisetti, *Internet of Things: A Hands-on Approach*, Universities Press.

Reference Books:

1. Raj Kamal, *Internet of Things: Architecture and Design*, McGraw Hill.
2. Pethuru Raj, Anupama Raman, *The Internet of Things: Enabling Technologies*, CRC Press.
3. Jan Holler et al., *From Machine-to-Machine to the Internet of Things*, Elsevier.
4. MQTT.org and CoAP documentation (IETF RFCs).
5. Cisco IoT Whitepapers & Protocol Stack Guidelines.

Track: Edge Computing (Honours / Double Minor)

SEM III	3KE245DH2	Introduction to Edge and Fog Computing	4	Theory
SEM IV	4KE246DH2	Edge Devices and Embedded Systems	4	Theory

Syllabus

Subject Code: 3KE245DH2
Subject Title: Introduction to Edge and Fog Computing

Semester: III
Credits: 4
Type: Theory

Course Pre-requisite:
Computer Fundamentals, Basics of Networking and Cloud Computing

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of edge and fog computing by being able to do each of the following:

- Understand the need for edge and fog computing as an extension of cloud computing.
- Explore architectures, key technologies, and components involved.
- Compare edge, fog, and cloud paradigms for latency-sensitive applications.
- Identify challenges, trends, and research directions.
- Analyze various use cases where edge/fog computing is applicable.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:

1. Explain the evolution from cloud to edge and fog computing.
2. Understand the architecture and components of edge and fog systems.
3. Evaluate the performance and trade-offs of edge/fog vs cloud computing.
4. Identify key technologies supporting edge and fog computing.
5. Apply edge/fog paradigms to real-world applications.
6. Analyze challenges and future directions in edge computing.

Syllabus:
Unit I: Fundamentals of Edge and Fog Computing (Hours: 07)
Introduction to distributed computing paradigms, Edge vs Cloud, Fog vs Edge, Importance of low-latency processing, Real-time analytics.

Unit II: Edge and Fog Architectures (Hours: 07)
Architectural models – layered and hierarchical, Cisco Fog computing model, OpenFog Consortium Reference Architecture, Edge gateways and micro data centers.
Unit III: Key Technologies and Platforms (Hours: 07)
IoT devices and sensors, Edge AI and ML, Virtualization (VMs, containers), Edge orchestration platforms (KubeEdge, EdgeX Foundry, Azure IoT Edge).
Unit IV: Communication Protocols and Standards (Hours: 07)
MQTT, CoAP, HTTP/HTTPS, BLE, Zigbee, 5G for edge, SDN/NFV in edge networks, Interoperability.
Unit V: Use Cases and Applications (Hours: 07)
Smart cities, Industrial IoT, Healthcare, Autonomous vehicles, Video surveillance, Retail and edge analytics.
Unit VI: Challenges and Research Trends (Hours: 07)
Security and privacy at the edge, Data management, Scalability, Energy efficiency, Standardization efforts, Emerging trends (serverless edge, federated learning).

Text Books:

1. Flavio Bonomi et al., *Fog Computing: A Platform for Internet of Things and Analytics*, Wiley.
2. Rajkumar Buyya, Satish Narayana Srirama, *Fog and Edge Computing: Principles and Paradigms*, Wiley.

Reference Books:

1. Tom Laszewski et al., *Cloud and Edge Computing*, Packt.
2. Perry Lea, *Internet of Things for Architects*, Packt.
3. Weisong Shi, Schahram Dustdar, *Edge Computing in the Internet of Things*, Springer.
4. Tutorials on EdgeX Foundry, Azure IoT Edge, KubeEdge (official docs).
5. IEEE papers and OpenFog Consortium whitepapers.

Subject Code: 4KE246DH2
Subject Title: Edge Devices and Embedded Systems

Semester: IV
Credits: 4
Type: Theory

Course Pre-requisite:
Fundamentals of IoT, Introduction to Edge and Fog Computing, Basics of Digital Electronics

Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of edge devices and embedded systems by being able to do each of the following:
• Understand the architecture and functioning of edge devices and embedded systems.
• Interface sensors and actuators with microcontrollers.
• Learn real-time data acquisition, processing, and transmission at the edge.
• Explore development environments and platforms for embedded applications.
• Build foundational skills for developing embedded and edge-enabled smart systems.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to:
1. Explain the role and structure of embedded systems in edge computing.
2. Interface hardware components and implement real-time edge solutions.
3. Develop software for microcontrollers and edge devices.

4. Utilize development platforms like Arduino, Raspberry Pi, ESP32.
5. Analyze performance constraints in embedded edge environments.
6. Design basic edge applications integrating embedded hardware and software.

Syllabus:

Unit I: Introduction to Embedded Systems and Edge Devices (Hours: 07)

Definition and characteristics of embedded systems, Real-time systems, Role in edge computing, Classification of edge devices (Arduino, Raspberry Pi, ESP8266, Jetson Nano).

Unit II: Microcontroller and Processor Architecture (Hours: 07)

Microcontrollers vs microprocessors, ARM architecture, GPIOs, Timers, ADC, Interrupts, Memory hierarchy in edge devices.

Unit III: Sensor and Actuator Interfaces (Hours: 07)

Analog and digital sensors, Interfacing methods (I2C, SPI, UART), Actuator control (motors, relays), Real-world sensing for edge analytics.

Unit IV: Embedded Programming and OS (Hours: 07)

Embedded C and Python basics, Programming microcontrollers (Arduino IDE, MicroPython), Lightweight OS for edge (RIOT, FreeRTOS, TinyOS).

Unit V: Communication and Connectivity (Hours: 07)

Wired and wireless interfaces, Bluetooth, Wi-Fi, LoRa, Zigbee, Integration with cloud/edge platforms (Node-RED, MQTT brokers, Firebase).

Unit VI: Applications and Case Studies (Hours: 07)

Use cases: Smart home, Smart grid, Industrial automation, Wearables, Hands-on project ideas using embedded systems and edge communication.

Text Books:

1. Raj Kamal, *Embedded Systems: Architecture, Programming and Design*, McGraw Hill.
2. John Catsoulis, *Designing Embedded Hardware*, O’Reilly.

Reference Books:

1. Muhammad Ali Mazidi, *AVR Microcontroller and Embedded Systems*, Pearson.
2. Lyla B. Das, *Embedded Systems – An Integrated Approach*, Pearson.
3. Jonathan Valvano, *Embedded Systems: Real-Time Interfacing*, CreateSpace.
4. Arduino & Raspberry Pi Official Guides and Community Documentation.
5. IEEE articles on IoT/Embedded & Edge Integration.