

MAHARSHI DAYANAND SARASWATI UNIVERSITY

AJMER



Syllabus

**Scheme of Examination and
Courses of Study**

Faculty of Science

Master of Technology (Computer Science)

Semester I and II w.e.f. 2026-27

Semester III and IV w.e.f. 2027-28

SCHEME OF TWO YEARS POST GRADUATE DEGREE PROGRAM

Master of Technology (Computer Science)

Program Code:

Program Faculty: Computer Science

Program Name: Master of Technology (Computer Science)

1. Introduction to the Program

The **Master of Technology (Computer Science)** is a premier two-year post-graduate degree program designed to bridge the gap between advanced academic computing research and high-demand industrial engineering workflows. In an era defined by explosive data growth, ubiquitous cloud architectures, and the transformative power of artificial intelligence, this curriculum offers an updated pathway through modern computer science specialization.

The program provides rigorous theoretical depth alongside heavy laboratory implementation. Students dive deep into core competencies including Information and Network Security, High-Performance Cloud Architecture, Advance Data Mining, and Machine Learning. The curriculum allows students to customize their academic trajectories through specialized tracking electives in Natural Language Processing, Deep Learning, Generative AI, and Locational Analytics using Spatial DBMS platforms. The program concludes with a comprehensive capstone dissertation, training graduates to spearhead technical innovations, publish peer-reviewed research, or architect complex enterprise software systems.

2. Objectives of the Program

The primary objectives of the M.Tech (Computer Science) program are to:

Excellence in Research & Innovation: To equip students with advanced cognitive and technical capabilities required to investigate complex computing anomalies and author high-impact scientific solutions.

Core Technological Expertise: To foster mastery over advanced systems engineering, cloud infrastructure deployment, cryptographic defense mechanisms, and parallelized processing architectures.

Data-Driven Problem Solving: To develop computational competencies in structuring, processing, and translating multi-structured big data, statistical vectors, and geographic data blocks into actionable business intelligence matrices.

Enterprise Software Automation: To cultivate capabilities in designing, validating, and managing large-scale application development loops utilizing agile workflows, strict software engineering patterns, and comprehensive QA metrics.

Leadership & Professional Ethics: To prepare graduates for high-level technical leadership, instilling a firm understanding of ethical AI governance, data privacy laws, intellectual property rights, and sustainable engineering values.

3. Employment and Entrepreneurial Scope

Employment Scope

Graduates of this program possess an advanced, highly specialized technical skillset that maps onto elite, high-value tech roles across the global industry:

AI & Machine Learning Engineer: Designing, optimizing, and deploying complex neural network models, deep learning frameworks, and generative AI systems.

Data Scientist & Big Data Architect: Building distributed analytical configurations, predictive models, and high-throughput data processing workflows using HDFS, NoSQL environments, and R/Python pipelines.

Cybersecurity Operations Director: Leading threat mitigation teams, configuring Zero Trust perimeters, and auditing advanced cryptographic infrastructure installations.

Enterprise Cloud & Systems Engineer: Architecting resilient, elastically scalable infrastructure setups (IaaS/PaaS/SaaS) on clusters, grids, and multi-tenant cloud grids.

Spatial Business Intelligence Analyst: Engineering location intelligence dashboards and managing enterprise spatial relational databases for strategic commercial market mapping.

Entrepreneurial Scope

The intense focus on practical toolchain mastery (such as Android SDK, JDeveloper, Oracle ADF, SSIS/SSAS, and modern LLM API integrations) enables graduates to pioneer digital transformation ventures:

SaaS Tech Startups: Launching niche, domain-specific Software-as-a-Service platforms backed by scalable cloud storage, automated pipelines, and custom web frontends.

Independent Boutique Consultancies: Establishing professional service agencies specializing in advanced corporate software testing automation, network vulnerability assessment, or data lake migrations.

Location Intelligence Venturing: Creating innovative consumer facing apps or business-to-business analytical modules leveraging GIS tracking networks and mobile application ecosystems.

AI Solution Integration: Building custom software agents or workflow plug-ins that fine-tune generative text/vision systems for domain-specific corporate clients.

4. Learning Outcomes of the Program (PO)

Upon successful completion of the M.Tech (Computer Science) program, graduates will achieve the following Program Outcomes (POs):

PO1 (Advanced Core Knowledge): Apply complex computing math, data structure heuristics, and cryptographic principles to engineer robust, high-performance computing components.

PO2 (Problem Analysis & Research): Identify, isolate, and formulate scientific solutions to complex multi-dimensional computer science research challenges by conducting exhaustive digital literature reviews.

PO3 (Design & Development of Solutions): Architect complete, end-to-end multi-tier software ecosystems, covering relational database persistence tiers, responsive mobile visual widgets, and intelligent application layers.

PO4 (Data Mining & Intelligence): Extract concealed trends, spatial hot-spots, and behavioral correlations from massive heterogeneous datasets using high-scale NoSQL nodes and statistical R/Python environments.

PO5 (Modern Tool Usage): Master and select proper development IDEs, container engines, machine learning suites, and software simulators (e.g., JDeveloper, Android SDK, ggplot2, Scikit-Learn) to build reliable technological solutions.

PO6 (Project Management & DevOps): Perform rigorous software lifecycle metric estimations (COCOMO cost lines, size metrics) and establish stable, automated quality assurance pipelines.

PO7 (Communication & Professionalism): Author technical reports, Software Requirements Specification (SRS) documentation, and peer-reviewed dissertation scripts while confidently delivering executive technical presentations.

SCHEME AND WEIGHTAGE OF COURSES

Semester I

S. No.	Course Category	Title of the Course	Credits			Credits earned	Evaluation			Remarks
			L	T	P		IA	ESA	Total	
SEMESTER – I										
1	mtc-101 DCC – 01	Information & Network Security	3	1	0	4	30	70	100	All Courses are compulsory Courses
2	mtc-102 DCC – 02	High-Performance Computing	3	1	0	4	30	70	100	
3	mtc-103 DCC – 03	Wireless Ad-hoc Network	3	1	0	4	30	70	100	
4	mtc-104 DCC – 04	Advance Data Mining	3	1	0	4	30	70	100	
5	mtc-105 DCC – 05	Lab - Information & Network Security and Advance Data Mining	0	0	4	4		100	100	
		Total				20			500	

Semester II

S. No.	Course Category	Title of the Course	Credits			Credits earned	Evaluation			Remarks
			L	T	P		IA	ESA	Total	
SEMESTER - II										
6	mtc-201 DCC – 06	Data Science with R	3	1	0	4	30	70	100	
7	mtc-202 DCC – 07	Java Application Development	3	1	0	4	30	70	100	
8	mtc-203 DCC – 08	Advance Computer Graphics	3	1	0	4	30	70	100	
9	mtc-204 a DSE/GEC– 01A	Advance Software Engineering	3	1	0	4	30	70	100	Student may opt any one
	mtc-204 b DSE/GEC – 01B	Machine Learning	3	1	0	4	30	70	100	
10	mtc-205 DCC – 09	Lab- Data Science with R and Java Application Development	0	0	4	4		100	100	
		Total				20			500	

Semester III

S. No.	Course Category	Title of the Course	Credits			Credits earned	Evaluation			Remarks
			L	T	P		IA	ESA	Total	
SEMESTER - III										
11	mtc-301 DCC – 10	Mobile Application Development	3	1	0	4	30	70	100	
12	mtc-302 DCC – 11	Advanced Image Processing	3	1	0	4	30	70	100	
13	mtc-303 a DSE – 02A	Big Data Analysis	3	1	0	4	30	70	100	Student may opt any one
	mtc-303 b DSE – 02B	Locational Analytics with Spatial DBMS					30	70		
14	mtc-304 a DSE – 03A	Software Testing	3	1	0	4	30	70	100	Student may opt any one
	mtc-304 b DSE – 03B	Natural Language Processing					30	70		
15	mtc-305 DCC – 12	Lab-Mobile Application Development and Advance Image Processing	0	0	4	4		100	100	
		Total				20			500	

Semester IV

S. No.	Course Category	Title of the Course	Credits			Credits earned	Evaluation			Remarks
			L	T	P		IA	ESA	Total	
SEMESTER - IV										
16	mtc-401 DCC – 13	Advanced Python for Data Science	3	1	0	4	30	70	100	
17	mtc-402 DCC – 14	Dissertation	3	1	0	4	30	70	100	
18	mtc-403 a DSE – 04A	Business Forecasting and Predictive Analytics	3	1	0	4	30	70	100	Student may opt any one
	mtc-403 b DSE – 04B	Business Intelligence using Locational Analytics					30	70		
19	mtc-404 a DSE – 05A	Object Oriented Software Engineering	3	1	0	4	30	70	100	Student may opt any one
	mtc-404 b DSE – 05B	Deep Learning and Generative AI					30	70		
20	mtc-405 DCC – 15	Lab Advanced Python for Data Science	0	0	4	4		100	100	
		Total				20			500	

Note: The evaluation of Dissertation shall be done on the basis of Dissertation Report and Viva voce Examination conducted by an external examiner and an internal examiner instead of semester end external exam.

Scheme of Evaluation

(A) Internal Evaluation – 30 Marks

The internal assessment shall consist of the following components: -

- First Internal Test – 10 Marks
- Second Internal Test – 10 Marks
- Hand written Assignment – 05 Marks
(to be written in approximately 500 words)
- Attendance – 05 Marks
- Total – 30 Marks**

The detailed records and supporting documents relating to Internal Assessment shall be maintained by the respective Departments/Colleges in accordance with the Standard Operating Procedure issued by the University Examination Section.

(B) End Semester Examination – 70 Marks

There shall be a Terminal Examination at the end of each Semester carrying 70 marks. The question paper shall comprise three sections **Section A, Section B and Section C.**

Section A

Section shall be of **20 marks** and consist of **10 questions** carrying **2 marks** each to be answered in approximately *20 words* each. All questions shall be compulsory. The examiner shall set the question paper choosing at least 3 questions from each unit.

Section B

Section shall be of **30 marks** and consist of **9 questions** carrying **5 marks** each to be answered in approximately *250 words* each. A Candidate shall attempt any six questions. The examiner shall set the questions ensuring that at least three questions are drawn from each Unit.

Section C

Section shall be of **20 marks** and consist of three questions carrying 10 marks each to be answered in *500 words* each. A Candidate shall attempt any two questions. The examiner shall set one question from each Unit.

Distribution of Marks Component: -

- **Internal Evaluation** – **30 marks**
- **End-Semester Examination** – **70 marks**

Note : –

- (i) The examiner shall ensure adequate coverage of the prescribed syllabus while setting the question paper and, maintain a balanced representation of all Units.
- (ii) The Teacher concerned, shall upload the marks online and submit awards in **3 Copies**, One will be retained by the Head / Principal and Two Copies of awards duly signed both by the Head/ Principal and the Teacher concerned shall be sent to the University Examination Section.
A Copy of the Proforma shall be sent every college/ department by the University Examination Section.
- (iii) The student has to qualify both the Internal Assessment and External Assessment separately. The minimum qualifying marks in both shall be 40 percent.
- (iv) The scheme will be applicable only to Major/ Multidisciplinary Subject (Not to AEC, SEC and VAC) opted by the student.

Scheme of Examination

Master of Technology (Computer Science)

Reg. 17 (a)

The M.Tech (Computer Science) Program will consist of 4 semesters. The examination shall consist of (a) Theory papers (b) Laboratory / Practical work and Dissertation work as per the scheme provided supra. Candidates will be required to pursue a regular, full time course of study at the College / University department for a period of two academic years in order to be eligible for appearing in the examination.

1. Eligibility for M.Tech Computer Science : M.Sc (CS)/M.Sc. (IT), M. Sc. (Physics), M. Sc. (Maths), MCA and B.E. / B. Tech. with minimum 60%.
2. Examination:
 - i. Scheme of Evaluation is provided supra.
 - ii. To pass a semester a candidate shall have to score 40% marks in each subject (theory and practical) separately and also 40% marks in aggregate of all the papers prescribed for the examination.
 - iii. Due paper(s) will be applicable if a candidate obtains 40% marks in aggregate and fails in not more than two (2) papers (theory). Due paper(s) of first semester will be held along with the third semester and the due paper(s) of second semester will be held along with the fourth semester. The third and fourth semester due paper(s) will be held in the first and second semester respectively of the next year. The chance of due paper(s) will be given only thrice in each semester.
 - iv. Wherever a candidate appears at for a due paper examination he/she will do so according to the syllabus in force.
 - v. A candidate not appearing at any examination/absent in any paper of term end examination shall be deemed as fail.
3. No Division will be awarded in the first, second and third semester examinations. Division shall be awarded at the end of the fourth semester Examination on the combined marks obtained at the first, second third and fourth semester taken together as noted below:

Passed with First Division	60% of the aggregate marks taken together of all the
	four semester examinations
Passed with second division	48%

All the rest will be declared to have passed the examination.
4. The grace marks shall be given up to 1% of the total aggregate marks of theory and practical of that semester in maximum one paper.
5. Candidates reappearing at an examination in a subsequent year shall be examined in accordance with the scheme and syllabi in force and shall be entitled to the award of the degree of year in which they clear the last failing/unclear paper.

mtc-101: Information & Network Security	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Core Principles: To master foundational concepts of data confidentiality, integrity, availability (CIA), and the core mathematical underpinnings of computer security.

Cryptographic Design: To analyze and implement the algorithmic mechanics of both traditional symmetric block ciphers and modern advanced encryption standards.

Asymmetric Infrastructure: To evaluate public-key cryptographic frameworks, mathematical algorithm design, and structured Public Key Infrastructure (PKI).

System & Network Defense: To engineer secure network perimeters using firewalls, intrusion prevention tools, access control models, and modern Zero Trust concepts.

Wireless & Mobile Protection: To appraise vulnerabilities unique to wireless ecosystems and mobile devices, developing robust mitigation policies for enterprise environments.

2. Course Outcomes (COs)

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

CO1: Differentiate between classical and modern symmetric cryptographic algorithms to assess their structural resilience against brute-force cryptanalysis.

CO2: Compute and validate mathematical encryption pipelines for symmetric standards (DES/AES) and asymmetric frameworks (RSA).

CO3: Design secure, scalable key distribution architectures utilizing decentralized controls, hybrid encryption, and formal certificate protocols.

CO4: Formulate technical mitigation strategies against modern wireless and mobile network vulnerabilities using industry-standard wireless defense policies.

CO5: Evaluate system perimeter defenses by configuring network security architectures, intrusion detection engines, and access control models.

3. Course Syllabus

mtc-101: Information and Network Security	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Symmetric cipher model, security attacks, services, and mechanisms. Cryptanalysis and brute-force attacks. Substitution techniques: Caesar, Monoalphabetic, Playfair, Hill, and Polyalphabetic ciphers. One-Time Pad (OTP). Block Ciphers & DES, Stream ciphers vs. block ciphers. Feistel cipher structure and design motivations. The Data Encryption Standard (DES): encryption, decryption, and avalanche effect. Strength and cryptanalysis of DES. Introduction to the Advanced Encryption Standard (AES) structure, byte substitution, shift rows, mix columns, and key expansion. Block cipher modes of operation (ECB, CBC, CFB, OFB, CTR).

Unit 2

Principles of public-key cryptosystems, applications, and mathematical requirements. The RSA algorithm: key generation, encryption, decryption, and security analysis. Brief introduction to Elliptic Curve Cryptography (ECC). Symmetric key distribution scenarios, hierarchical key control, session key lifetimes, and decentralized key control. Controlling key usage. Symmetric key distribution using asymmetric encryption, simple secret key distribution, and secret key distribution with confidentiality and authentication. Hybrid cryptographic schemes. Distribution of public keys and Public Key Infrastructure (PKI).

Unit 3

Wireless Network Security, Wireless security protocols (WEP, WPA2, WPA3). Wireless network threats (eavesdropping, jamming, rogue access points) and defensive engineering countermeasures. Mobile & Device Security, Mobile ecosystem threats, malware, and data leakage risks. Mobile Device Management (MDM) frameworks and containerization strategies. Firewalls, Intrusion Detection/Prevention Systems (IDS/IPS). Conceptual overview of Zero Trust Architecture (ZTA) and secure remote access (VPNs, TLS/SSL)

4. Suggested Books and References

- [1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 8th ed. Upper Saddle River, NJ, USA: Pearson, 2020.
- [2] B. A. Forouzan and D. Mukhopadhyay, *Cryptography and Network Security*, 3rd ed. New Delhi, India: McGraw Hill Education, 2015.
- [3] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 20th anniversary ed. Hoboken, NJ, USA: Wiley, 2015.
- [4] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed. Boston, MA, USA: Prentice Hall, 2002.
- [5] *Computer Security Resource Center*, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 2026. [Online]. Available: nist.gov

mtc-102: High-Performance Computing	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Distributed Foundations: To evaluate the architectural transitions from legacy distributed systems and cluster configurations to high-performance Grid networks.

Grid Services & Standards: To master structural grid concepts, resource sharing protocols, and service frameworks including OGSA, WSRF, and the Semantic Grid.

Cloud Core Mechanics: To analyze the NIST cloud model, deployment modes, and underlying multi-tenant hypervisor virtualization architectures.

Service Plane Delivery: To deploy scalable corporate infrastructures using standard cloud delivery mechanisms including IaaS, PaaS, and SaaS ecosystems.

SLA & Cloud Security: To engineer secure cloud boundaries managed by automated Service Level Agreements (SLAs), rigorous identity management, and data protection rules.

2. Course Outcomes (COs)

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

CO1: Differentiate between distributed computing topologies by grading performance metrics across cluster, grid, and cloud setups.

CO2: Build metadata service registries and automated node workflows governed by OGSA, WSRF, and GSI specifications.

CO3: Configure isolated virtual networks, custom machine templates, and secure cloud storage targets using standard hypervisors.

CO4: Model commercial software applications across appropriate SaaS, PaaS, or IaaS delivery layers based on enterprise requirements.

CO5: Enforce compliance scripts, identity access matrices, and automated SLA monitors to defend cloud pipelines against resource misuse.

3. Course Syllabus

mtc-102: High-Performance Computing	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Evolution of distributed paradigms: Cluster, Grid, and Utility computing models. Introduction to Grid Architecture, characteristics, standard regulatory bodies, grid types, and topological designs. Architectural differences between computational grids and data grids. Open Grid Services Architecture (OGSA) and Web Services Resource Framework (WSRF). The Semantic Grid, Metadata, ontologies, and layered structures..

Unit 2

Cloud Computing Architecture, Service Models, Virtualization, Cloud Principles, Definition and core characteristics. Cloud computing stack, Network protocols, web interfaces, and routing mechanisms in data centers. Deployment models: Public, Private, Hybrid, and Community clouds. Introduction to Infrastructure as a Service (IaaS). Foundations of virtualization, Type-1 and Type-2 hypervisors, and machine image building blocks. Multi-tenant provisioning of virtual machines (VMs), CPU slicing, virtual networking, and Storage as a Service (STaaS).

Unit 3

Platform as a Service (PaaS) architecture and Service-Oriented Architecture (SOA) models. Application runtime environments (e.g., Google App Engine, Microsoft Azure). Software as a Service (SaaS) mechanics, Web 2.0 paradigms, and Web Operating Systems. Service Level Agreements (SLAs) lifecycle management, metered billing frameworks, and cloud accounting tools. Economics of elastic vertical and horizontal scaling.

Multi-layered security boundaries: Network, host, and application planes. Data privacy mechanisms, storage threats, and cross-tenant isolation enforcement. Identity and Access Management (IAM), role-based access control (RBAC), and authentication. Legal considerations, business risk modeling, and cloud contracting requirements.

4. Suggested Books and References

- **[1]** K. Hwang, G. C. Fox, and J. J. Dongarra, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, 1st ed. Boston, MA, USA: Morgan Kaufmann Publishers, 2011.
- **[2]** I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2004.
- **[3]** B. Sosinsky, *Cloud Computing Bible*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
- **[4]** R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, 1st ed. New York, NY, USA: McGraw-Hill Education, 2013.
- **[5]** P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, Special Publication 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Sep. 2011. [Online]. Available: nist.gov

mtc-103: Wireless Ad-hoc Network	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

Physical & Medium Foundations: To master fundamental wireless physical layer concepts, multiple access technologies, and radio propagation channel modeling.

Local Area Networks: To analyze the protocol architectures, MAC sublayer management mechanisms, and deployment challenges of IEEE 802.11 WLAN ecosystems.

Cellular Evolution: To evaluate structural transitions across generations of cellular networks (1G to 4G) alongside core infrastructure mobile routing protocols.

Ad-hoc & Sensor Systems: To design resource-constrained Wireless Sensor Networks (WSNs) and Wireless Personal Area Networks (WPANs) incorporating power optimization techniques.

Wireless Security Defenses: To engineer robust mitigation countermeasures protecting mobile and wireless networks against perimeter vulnerabilities and Denial of Service (DoS) attacks.

2. Course Outcomes (COs)

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

CO1: Calculate link budget parameters and evaluate capacity constraints across CDMA, TDMA, and FDMA wireless environments.

CO2: Configure and resolve resource contention conflicts in IEEE 802.11 WLANs by isolating hidden and exposed node anomalies.

CO3: Architect handoff matrices and channel assignment rules to optimize coverage within 3G/4G cellular core systems.

CO4: Deploy localized low-power sensor nodes managed by optimized routing, power-cycling, and TinyOS application scripts.

CO5: Build secure wireless perimeters by implementing advanced cryptographic authentication rules and mitigating DoS network vectors.

3. Course Syllabus

mtc-103: Wireless Ad-hoc Network	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Evolution of wireless networking trends. Core physical layer parameters, radio wave propagation, and channel fading modelling. Multiple access technologies: Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and spread spectrum techniques. Infrastructure networks vs. mobile ad-hoc networks (MANETs). Algorithmic resolution of the Hidden Node and Exposed Terminal problems via RTS/CTS handshaking. Mitigation of multipath fading effects in indoor and outdoor WLANs. Enterprise WLAN deployment engineering and coverage optimization issues.

Unit 2

Architectural development of cellular standards across 1G, 2G, 2.5G, 3G, and 4G platforms. Cellular design parameters: Frequency reuse factors, fixed vs. dynamic channel assignment strategies, and soft/hard handoff strategies. Management of co-channel and adjacent-channel interference to optimize system capacity. Engineering methodologies for improving coverage (cell splitting, sectoring, and microcell zones).

Routing paradigms in IP mobility: Mobile IPv4 and Mobile IPv6 architectural frameworks, registration pipelines, and tunneling mechanisms. Worldwide Interoperability for Microwave Access (WiMAX) frameworks including physical layer properties, media access control scheduling, mobility, and network layout.

Unit 3

Wireless Sensor Networks (WSNs), architecture of WSN, industrial applications. Multi-layered protocol design across WSN Physical, MAC (S-MAC, T-MAC), and Network routing layers (directed diffusion, clustering protocols. Short-range communication topologies including Bluetooth (IEEE 802.15.1) piconets/scatternets and Zigbee (IEEE 802.15.4) mesh structures, Wi-Fi Protected Access (WPA/WPA2/WPA3), and IEEE 802.11i architectures.

4. Suggested Books and References

- [1] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [2] J. Schiller, *Mobile Communications*, 2nd ed. London, UK: Pearson Education, 2003.
- [3] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2005.
- [4] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
- [5] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2004.

mtc-104: Advance Data Mining	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

Foundations & Preparation: To understand the evolution, trends, and lifecycle of data mining while mastering data preprocessing pipeline mechanisms.

Warehouse Modeling: To analyze data warehouse architectures, multidimensional data cubes, online analytical processing (OLAP), and structural extract-transform-load (ETL) workflows.

Pattern & Rule Discovery: To apply algorithmic frameworks for discovering frequent itemsets, mining association rules, and extracting correlation structures from complex data.

Predictive Analytics: To evaluate the mathematical mechanics of predictive modeling, classification engines, error estimation metrics, and cluster analysis.

Enterprise Intelligence: To engineer production-ready business intelligence architectures using key performance indicator (KPI) structures, dashboards, and enterprise tools.

2. Course Outcomes (COs)

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

CO1: Build automated data preprocessing pipelines to clean, integrate, transform, and discretize messy transactional datasets.

CO2: Design multidimensional data warehouse models and execute high-performance OLAP cube operations for complex business analysis frameworks.

CO3: Implement frequent itemset extraction and association rule mining algorithms to uncover hidden transactional correlations.

CO4: Deploy optimized classification, prediction, and cluster analysis models while verifying system accuracy using statistical error metrics.

CO5: Architect enterprise business intelligence environments by building interactive KPI dashboards and data integration workflows using SSAS, SSIS, and SSRS suites.

3. Course Syllabus

mtc-104: Advance Data Mining	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Evolution and importance of data mining, Data cleaning methodologies for missing or noisy attributes. Data integration protocols and redundancy analysis. Data reduction techniques, data transformation mapping, and continuous attribute discretization, Concepts of enterprise data repositories. Data warehouse modeling frameworks: star, snowflake, and fact constellation schemas. Comparative analysis of Online Analytical Processing (OLAP) and Online Transaction Processing (OLTP) systems. Data cube construction, computation, and multidimensional OLAP operations (slice, dice, roll-up, drill-down).

Unit 2

Frequent Itemset Mining, market basket analysis. Algorithmic methods for frequent itemset extraction (Apriori and Frequent Pattern Growth paradigms). Mining advanced association rules from transactional datasets.

Classification & Predictive Modeling, supervised machine learning. Decision tree induction algorithms and information gain criteria. Bayesian classification mechanics and Naive Bayes architectures. Rule-based classification engines, Linear and non-linear regression prediction models. Performance validation using accuracy, error measures, confusion matrices, and ROC curves.

Unit 3:

Cluster Analysis, Unsupervised learning overview and similarity-distance metrics. Structural analysis of data types in cluster systems. Major clustering methodologies: partitioning methods, hierarchical methods, and density-based spaces (DBSCAN). Web mining, temporal and spatial data mining

BI Architecture concepts, and core layers of the Business Intelligence framework. Operational basics of data integration, data staging, and automated data pipeline management. Core design concepts of executive dashboards and balanced scorecard management structures.

4. Suggested Books and References

- [1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan Kaufmann Publishers, 2011.
- [2] E. Turban, R. Sharda, and D. Delen, *Business Intelligence and Analytics: Systems for Decision Support*, 10th ed. Upper Saddle River, NJ, USA: Pearson, 2014.
- [3] A. Berson and S. J. Smith, *Data Warehousing, Data Mining, and OLAP*, 1st ed. New York, NY, USA: McGraw-Hill Education, 1997.
- [4] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 2nd ed. Boston, MA, USA: Pearson, 2018.
- [5] C. M. Saracco, D. Balsamon, and G. S. Deep, *Microsoft SQL Server Analysis Services, Integration Services, and Reporting Services: Practical Guide for BI Developers*, 1st ed. Redmond, WA, USA: Microsoft Press, 2016.

mtc-201 Data Science with R	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

To master the core syntax, data structures, control statements, and file I/O operations using the R environment.

To engineer high-quality statistical data visualizations utilizing base R graphics and advanced ggplot2 frameworks.

To evaluate dataset characteristics through data sampling techniques, confidence intervals, and parametric/non-parametric hypothesis testing.

To design, build, and evaluate robust predictive models using advanced regression and classification algorithms.

To apply feature engineering, dimensionality reduction techniques, clustering algorithms, and transactional association rules.

2. Course Outcomes (COs)

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

CO1: Write structured, efficient R scripts utilizing appropriate vector, matrix, list, and data frame manipulation pipelines.

CO2: Generate publication-ready data visualizations by applying the Grammar of Graphics with the ggplot2 framework.

CO3: Conduct parametric and non-parametric hypothesis tests in R to draw mathematically sound inferences from sample datasets.

CO4: Architect and validate linear, logistic, and non-linear regression models optimized through cross-validation techniques.

CO5: Deploy advanced machine learning models for classification, dimensionality reduction, clustering, and association rule mining using standard R libraries.

3. Course Syllabus

mtc-201 Data Science with R	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Introduction to the R workspace, command-line interface, and basic syntax elements. Data objects, type systems, structural attributes, numbers, vectors, arrays, matrices, data frames, and lists. Control Flows & Scripting: Reading data from external flat files, databases, and structural datasets. Conditional control statements, nested loops, vectorization, custom functions, scoping rules, and writing executable R scripts. Overview of the data science lifecycle. Advanced Data Visualization: Techniques for exploratory data analysis. Static rendering using base R graphics engines.

Unit 2

Statistical Inference Foundations: Introduction to statistical hypotheses, null vs. alternative hypotheses, Type I and Type II errors. Data sampling methodologies, sample size determination, confidence intervals, and statistical significance levels. Hypothesis Testing Frameworks: Parametric testing execution in R (one-sample/two-sample t-tests, paired t-tests, ANOVA). Non-parametric testing models (Wilcoxon signed-rank test, Mann-Whitney test, Chi-square test of independence). Regression Modelling, continuous prediction and regression analysis. Ordinary Least Squares (OLS) linear regression.

Unit 3:

Dimensionality Reduction & Feature Selection, Principal Component Analysis (PCA) mechanics, Supervised pattern recognition. Logistic regression modelling for binary and multi-class outcomes. R implementation of Support Vector Machines (SVM) and Naive Bayes probabilistic classifiers. Decision tree and Random Forest classification. Methods for evaluating classifier performance (Confusion Matrix, Precision, Recall, F1-Score, ROC-AUC curves). Unsupervised Clustering & Association Rules: Introduction to cluster analysis and distance metrics. Partitioning and hierarchical clustering methods k-means. Market basket analysis and frequent itemset extraction using association rules via the Apriori algorithm.

4. Suggested Books and References

- [1] H. Wickham and G. Grolemund, *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2023.
- [2] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd ed. New York, NY, USA: Springer, 2021.
- [3] N. Zumel and J. Mount, *Practical Data Science with R*, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2019.
- [4] J. Maindonald and J. Braun, *Data Analysis and Graphics Using R: An Example-Based Approach*, 3rd ed. Cambridge, UK: Cambridge University Press, 2010.
- [5] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2026. [Online]. Available: r-project.org

mtc-202	Duration: 3 hours
Java Application Development	Max Marks: 70

1. Course Objectives

The primary objectives of this course are to:

IDE & Middleware Foundations: To master the JDeveloper workspace ecosystem, integrated testing containers, and deployment architectures on Oracle WebLogic Server.

Business Logic Layering: To engineer robust, transactional business logic layers using Oracle ADF Business Components (BC) including Entity Objects and View Objects.

User Interface Design: To build dynamic, enterprise-grade server-side web interfaces using ADF Faces rich client components and task flow bounded logic.

Data Binding & Security: To implement the MVC model via the ADF Model binding layer, declarative validation rules, and transactional container security.

Enterprise Service Integration: To expose backend application modules as industry-standard SOAP and RESTful web services using JDeveloper integration wizards.

2. Course Outcomes (COs)

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

CO1: Configure and manage the JDeveloper IDE environment alongside local and remote WebLogic application server instances.

CO2: Build transactional business services by designing performance-optimized ADF Entity Objects, View Objects, and Application Modules.

CO3: Implement scalable application navigation matrices by configuring bounded task flows, contextual parameters, and routers.

CO4: Design enterprise-grade web frontends by deploying multi-layered ADF Faces rich client interfaces with automated data binding.

CO5: Secure web boundaries and expose business data payloads to external clients by building authenticated REST/SOAP endpoints.

3. Course Syllabus

mtc-202 Java Application Development	Duration: 3 hours Max Marks: 70
---	--

Unit 1

JDeveloper Environment, The JDeveloper Ecosystem, Workspace anatomy, project structures, and configuration files. Customizing the developer environment, utilizing structural wizards, and integrating source control. Architecture of the Oracle Application Development Framework (ADF) and its alignment with Model-View-Controller (MVC) paradigms. Designing Entity Objects (EOs) for database mapping, handling associations, and enforcing declarative business validation rules. Creating View Objects (VOs) for SQL abstraction, implementing View Links for master-detail relationships, and tuning query execution for performance.

Unit 2

Defining the data model, exposing custom client service methods, and optimizing application module pooling settings for multi-user enterprise loads. The Binding Layer (Controller) Understanding data controls, binding definitions (PageDef), and attribute/action bindings. Synchronizing the frontend view with backend business services.

Unit 3

Building rich internet applications using ADF Faces components. Implementing structural layouts, data tables, master-detail tree structures, popups, and advanced input components. Managing user state through partial page rendering (PPR) and managing managed beans for UI-specific logic.

Exposing ADF Application Modules as web services. Generating SOAP services using JDeveloper wizards and publishing RESTful endpoints with JSON payloads. Best practices for testing and packaging applications into enterprise archive (EAR) deployment files.

4. Suggested Books and References

- [1] D. Duncan, *Oracle ADF Real World Developer's Guide*, 1st ed. Birmingham, UK: Packt Publishing, 2012.
- [2] S. King, *Oracle JDeveloper 12c Handbook: A Guide to Fusion Web Development*, 1st ed. New York, NY, USA: McGraw-Hill Education, 2014.
- [3] R. Cardwell, *Developing Web Applications with Oracle ADF Essentials*, 1st ed. Birmingham, UK: Packt Publishing, 2013.
- [4] F. Nimphius and L. Munsinger, *Oracle Fusion Developer Guide: Building Rich Internet Applications with Oracle ADF Business Components and Oracle ADF Faces*, 1st ed. New York, NY, USA: McGraw-Hill Education, 2010.
- [5] *Oracle Fusion Middleware Developer's Guide for Oracle Application Development Framework*, Oracle Corporation, Austin, TX, USA, 2025. [Online]. Available: oracle.com

mtc-203 Advance Computer Graphics	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Geometric Modeling Complexity: To master mathematical and algorithmic representations of three-dimensional spaces using boundary polyhedra, implicit blobby objects, and parametric spline spaces.

Spatial Visibility Processing: To analyze, implement, and compare runtime performance benchmarks across image-space and object-space visible-surface detection algorithms.

Color Science Engineering: To evaluate the underlying physics of light, standard chromaticity curves, and hardware/software color model conversions across design frameworks.

Graphics API Synthesis: To implement production-ready 3D rendering scripts utilizing modern OpenGL shader pipelines and quadric/cubic API function wrappers.

Animation Foundations: To apply spatial kinematics, keyframing interpolation vectors, and behavioral rules to engineer fluid computer animation pipelines.

2. Course Outcomes (COs)

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

CO1: Formulate and program complex 3D geometric pipelines using OpenGL primitives, procedural blobby objects, and parametric spline surfaces.

CO2: Evaluate spatial complexity constraints to choose, implement, and optimize the ideal visible-surface detection algorithm for dense 3D scenes.

CO3: Build automated spatial acceleration structures like BSP-trees and Octrees to accelerate real-time ray intersection and frame culling operations.

CO4: Design programmatic color space conversion scripts to map visual assets accurately between RGB, YIQ, CMYK, and HSV engines.

CO5: Author mathematically fluid computer animations by integrating parametric easing trajectories and keyframe interpolation algorithms.

3. Course Syllabus

mtc-203 Advance Computer Graphics	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Three-Dimensional Object Representation, Spline Modeling, Polygonal & Quadratic Structures: Boundary representations, spatial data structures, and OpenGL Polyhedron functions.

Analytical rendering of curved surfaces: Quadric surfaces, superquadrics, and implicit blobby objects (metaballs). Application of OpenGL Quadric-surface and Cubic-surface primitives. Advanced Curve & Surface Design, Deep formulation of Bézier Spline curves and spatial Bézier surfaces.

Unit 2

Visible-Surface Detection Methods, Theoretical division between object-space and image-space culling methods. Buffer & Raster Scanline Algorithms: High-speed hardware rendering layouts: Depth-Buffer (Z-buffer) execution and the multi-layer A-Buffer anti-aliasing framework. Scan-Line visibility resolution, item buffer logic, and edge-table configurations. Spatial Partitioning, Ray Traversal, Recursive spatial hierarchy generation using Binary Space Partitioning (BSP-Trees), Area-Subdivision heuristics, and sparse Octree data spaces. Mechanics of Ray-Casting intersection tests. Comparative analysis of visibility optimization pipelines across wire-frame, polygon, and curved-surface geometries.

Unit 3:

Color Models, Chromaticity Science, Properties of electromagnetic light radiation, standard industrial primaries, and the CIE Chromaticity diagram. Mathematical transformation matrices across hardware, broadcast, and print color pipelines, The RGB model, YIQ / YCbCr broadcast variants, and the subtractive CMY/CMYK printing matrices. Computer Animation Mechanics, Kinematic path generation, keyframe parameter interpolation (linear vs. splined easing), and rigid-body mechanics. Real-time rendering deployment strategies inside standard modern graphics rendering loops.

4. Suggested Books and References

- [1] D. Hearn, M. P. Baker, and W. R. Carithers, *Computer Graphics with OpenGL*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2011.
- [2] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, 3rd ed. Boston, MA, USA: Addison-Wesley Professional, 2013.
- [3] E. Angel and D. Shreiner, *Interactive Computer Graphics: A Top-Down Approach with WebGL*, 7th ed. Boston, MA, USA: Pearson, 2014.
- [4] Tomas Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering*, 4th ed. Boca Raton, FL, USA: CRC Press, 2018.
- [5] Khronos Group, *OpenGL Registry & Modern Shader Reference*, Beaverton, OR, USA, 2026. [Online]. Available: [khronos.org](https://www.khronos.org)

mtc-204a Advance Software Engineering	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Understand the structural architecture of software processes, contrasting prescriptive lifecycles with modern, iterative agile development models (Scrum, XP, Scrum variants).

Understand the engineering principles of software requirements extraction, formal use case modeling, and the construction of standardized Software Requirements Specification (SRS) documents.

Apply architectural software design metrics, including module abstractions, information hiding, and optimization of functional independence via tight cohesion and loose coupling.

Examine quantitative mathematical management models for software project sizing (LOC/FP), cost estimation (COCOMO), scheduling pipelines, and proactive risk engineering (RMMM).

Implement comprehensive structural validation and verification strategies using automated white-box path testing, black-box behavioral testing, and structured change management protocols.

2. Course Outcomes (COs)

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

CO-1: Analyze and select appropriate software process methodologies (prescriptive vs. agile frameworks) to match specific organizational constraints, delivery speeds, and scale complexities.

CO-2: Construct comprehensive, unambiguous Software Requirements Specification (SRS) artifacts by eliciting user profiles, managing use cases, and leading peer reviews.

CO-3: Design modular, object-oriented software architectures by evaluating design patterns and optimizing functional coupling and cohesion properties.

CO-4: Formulate algorithmic estimations of software project size, engineering effort, financial costs, and operational timelines using COCOMO frameworks and Gantt scheduling networks.

CO-5: Structure end-to-end testing pipelines—including unit, basis path, integration, and regression matrices—while managing code evolution via Change Control Boards and automated configuration repositories.

3. Course Syllabus

mtc-204a Advance Software Engineering	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Software Processes & Characteristics, Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models, Overview of Quality Standards like ISO9001, SEI – CMM. Requirement engineering, requirement elicitation techniques like FAST, QFD & Use case approach, requirements analysis using DFD, ER Diagrams, Requirements documentation, Nature of SRS, Characteristics & organization of SRS. Project Management, Component Based Development. Agile Process Models – Extreme Programming (XP), Adaptive Software Development, Scrum, Dynamic System Development Model, Feature Driven Development, Crystal.

Unit 2

Software Project Planning: Size Estimation like lines of Code & Function Count, Cost Estimation Models, Static single & Multivariable Models, COCOMO, COCOMO-II, Putnam resource allocation model, Risk Management Software Design: Abstraction, Architecture, Patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Cohesion and Coupling; Object-Oriented Design, Data Design, Architectural Design, User Interface Design, Component Level Design. Software Quality: McCall's Quality Factors, ISO 9126 Quality Factors, Quality Control, Quality Assurance, Software Reliability.

Unit 3

Software Testing: Verification and Validation; Testing process, Design of test cases, functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path Testing, Data flow and mutation testing, Unit Testing, Integration and System Testing, Debugging, Alpha & Beta Testing, Regression Testing, Testing Tools & Standards.

4. Suggested Books and References

- [1] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill Education, 2020.
- [2] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.
- [3] K. K. Aggarwal and Y. Singh, *Software Engineering*, 3rd ed. New Delhi, India: New Age International Publishers, 2008.
- [4] P. Jalote, *An Integrated Approach to Software Engineering*, 3rd ed. New York, NY, USA: Springer, 2005.
- [5] IEEE Computer Society, *Software Engineering Body of Knowledge (SWEBOOK v4.0)*. [Online]. Available: computer.org
- [6] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th ed. Newtown Square, PA, USA: Project Management Institute, 2021.

mtc-204b Machine Learning	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Understand the core terminology, classification taxonomies, and foundational lifecycle steps of a standard machine learning pipeline.

Understand data engineering pre-processing methods, including missing data interpolation, categorical dimension encoding, feature scaling, and isolated validation splits

Apply linear and logistic regression models to solve continuous target estimation challenges and discrete binary classification problems.

Implement non-parametric machine learning methodologies, including entropy-based decision tree splitting paths and instance-based proximity voting.

Evaluate unsupervised data spatial groupings and multi-variate feature space reductions to extract hidden structures from unlabelled matrices.

2. Course Outcomes (COs)

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

CO-1: Construct structured data ingestion scripts that balance feature scales, parse categorical elements, and partition evaluation datasets safely.

CO-2: Design and optimize continuous target predictors using ordinary least squares and gradient descent frameworks while assessing bias-variance trade-offs.

CO-3: Formulate discrete categorical classification loops using logistic functions and decision trees, diagnosing performance via precision, recall, and ROC-AUC metrics.

CO-4: Model automated cluster configurations using partition-based K-Means and agglomerative linkage structures, verifying densities via silhouette profiles.

CO-5: Assemble matrix transformations using Principal Component Analysis (PCA) to compress high-dimensional feature systems into visual spaces.

3. Course Syllabus

mtc-204b Machine Learning	Duration: 3 hours Max Marks: 70
--	--

Unit 1:

Introduction to Machine Learning: Defining ML taxonomy; Supervised learning, Unsupervised learning, and Reinforcement learning paradigms; applications and ethical considerations. Data Preprocessing Pipeline, dataset partitioning (Train-Test splits). Linear Regression Analysis, Model Assessment Fundamentals, Understanding the bias-variance tradeoff; overfitting and underfitting; validation techniques.

Unit 2

Supervised Classification, Model Diagnostics, Categorical Classification Logic, Transitioning from continuous targets to discrete labels; understanding boundary margins and probability modeling. Logistic Regression Architecture, Binary classification workflows; activation function, and maximum likelihood estimation.

Introduction to Decision Trees; structural splitting metrics (Gini Impurity, Information Gain, Entropy), node pruning, and interpreting root-to-leaf decision pathways. Instance-Based Learning: Nearest Neighbor algorithms (k-NN); distance calculations (Euclidean, Manhattan metrics), choosing parameter k, and voting boundaries. Classification Diagnostics Matrix Confusion Matrix, Precision, Recall, F1-Score, ROC-AUC curves.

Unit 3

Unsupervised Learning & Clustering Architectures, un-labelled data patterns; understanding clustering vs. classification and data density structures. Partition-Based Clustering, The K-Means Clustering algorithm; Hierarchical Agglomerative Clustering, cluster tree tree-cutting configurations. Dimensionality Reduction Basics, Principal Component Analysis (PCA) theory, eigenvalue/eigenvector abstractions, and projecting high-dimensional datasets into 2D/3D spaces for visualization.

4. Suggested Books and References

- [1] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [2] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd ed. New York, NY, USA: Springer, 2021.
- [3] S. Marsland, *Machine Learning: An Algorithmic Perspective*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2014.
- [4] T. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill Education, 1997.
- [5] E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge, MA, USA: MIT Press, 2020.
- [6] Scikit-Learn Developers. *Scikit-Learn User Guide: Supervised and Unsupervised Workflows*. [Online]. Available: scikit-learn.org

mtc-301 Mobile Application Development	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

Understand the core principles of mobile computing architecture, including network protocols, mobile services, and handing-over mechanisms.

Deconstruct the multi-tier Android software stack and Understand the use of essential development toolchains like the Android SDK, Android Emulator, and the Android Debug Bridge (ADB).

Apply structural application components—such as Activities, Fragments, and event-handling frameworks—to engineer highly responsive graphical user interfaces.

Implement advanced interface enhancements, including adaptive resource allocation, custom canvas drawing, multi-stage dialog windows, and keyframe animations.

Develop secure, data-driven applications that interface seamlessly with external web services via WebViews/Maps, communicate through background services, and securely share data structures using native Content Providers.

2. Course Outcomes (COs)

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

CO-1: Analyze mobile computing infrastructures and map application requirements directly to the core runtime layers of the Android system architecture.

CO-2: Construct stable application environments utilizing complex UI widget matrices, explicit event handlers, and robust container layouts to handle platform mutations.

CO-3: Design modular, high-density presentation layers by building adaptive fragments, embedding multi-resolution media assets, and orchestrating interactive alert dialog pipelines.

CO-4: Formulate asynchronous network calls to securely display dynamic web pages, plot geo-spatial mapping elements, and manage low-level background threads away from main user processing loops.

CO-5: Assemble production-grade Android database systems using custom Content Providers and output an obfuscated, signed, release-ready binary build package for distribution.

3. Course Syllabus

mtc-301 Mobile Application Development	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Introduction to the concepts of mobile computing networks; identifying novel applications, structural mobile services, system architecture frameworks, handover mechanisms, and foundational network protocols.

Android Runtime Ecosystem: Introduction to the Android platform; unpacking the components of the Android SDK, parsing the layers of the multi-tier Android Software Stack, environment configurations, and initializing Android Virtual Devices (AVD). Constructing a primary project; utilizing the command-line Android Debug Bridge (ADB) client, manipulating the Android Emulator environment, and working with basic Text View controls.

Unit 2

Lifecycle and structural role of Android Application Components (Activities, Services, Broadcast Receivers, Content Providers); managing application layouts inside standard structural containers, and processing localized interaction events. Handling text variables via Edit Text controls, launching operational workflows using basic buttons/widgets, creating and starting an explicit Activity. Engineering flexible visual viewports; laying out controls inside specialized container objects, mapping adaptive layout matrices, and handling asset files cleanly within localized system hierarchies.

Unit 3

Implementing advanced user-selection matrices; configuring data-binding adapter views (List Views, Grid Layouts, Spinners) and executing code debugging routines. Modular design of interfaces; building reusable layout logic layers using Fragments, handling multi-stage operations using Dialog screens, and passing complex transactional arguments. Enhancing application user engagement metrics; drafting custom canvas drawing classes and implementing structural layer-based animations.

4. Suggested Books and References

- [1] B. Phillips, C. Stewart, and K. Marsicano, *Android Programming: The Big Nerd Ranch Guide*, 4th ed. Atlanta, GA, USA: Big Nerd Ranch Guides, 2019.
- [2] J. Horton, *Android Programming for Beginners*, 3rd ed. Birmingham, UK: Packt Publishing, 2021.
- [3] R. Hanselman and L. Blake, *Mobile Computing Foundations and Protocol Architectures*, 1st ed. New York, NY, USA: McGraw-Hill Education, 2017.
- [4] Google Developer Team. *Android App Architecture and Core Developer Guides*. [Online]. Available: android.com
- [5] W. Stallings, *Wireless Communications & Networks*, 2nd ed. Boston, MA, USA: Pearson Education, 2004.
- [6] Z. Mednieks, L. Dornin, G. Meike, and M. Nakamura, *Programming Android: Java Programming for the New Generation of Mobile Devices*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2012

mtc-302 Advanced Image Processing	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Perceptual Data Modeling: To master the physiological fundamentals of human visual perception, continuous coordinate sampling, and grayscale quantization frameworks.

Spatial Transformation Engineering: To develop advanced skills in spatial domain image enhancement, analytical histogram processing, and linear/nonlinear neighborhood filtering.

Frequency Domain Analysis: To analyze image characteristics in the frequency spectrum using the Discrete Fourier Transform alongside high-pass and low-pass filters.

Degradation & Reconstruction Optimization: To mathematically model structural image degradation and implement statistical noise filters and inverse recovery systems.

Color Informatics & Compression: To evaluate multi-channel color space models, chromatic segmentation, and data reduction strategies for digital transmission.

2. Course Outcomes (COs)

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

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

CO1: Process raw multidimensional datasets by executing pixel relationship measurements, grid interpolation resizing, and resolution scaling.

CO2: Build high-contrast image enhancement modules by designing programmatic gray-level transformations and 2D spatial convolution masks.

CO3: Translate spatial graphics into the frequency domain via 2D Fourier Transforms to apply smoothing and sharpening transfer functions.

CO4: Synthesize mathematical degradation matrices to reconstruct degraded images using spatial noise-reduction and inverse filtering.

CO5: Architect full-color processing pipelines to handle chromatic transformations, vector-space distance segmentations, and compression algorithms.

3 Course Syllabus

mtc-302 Advanced Image Processing	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Elements of Visual Perception, A Simple Image Formation Model, Basic Concepts in Sampling and Quantization, Representing Digital Images, Spatial and Gray-level Resolution, Zooming and Shrinking Digital Images, Algorithmic execution of digital image resizing: nearest-neighbor, bilinear, and bicubic zooming and shrinking frameworks. Basic spatial relationships between pixels: adjacency metrics (4, 8, and m-adjacency), path links, coordinate distances (Euclidean, City-Block, Chessboard), and connectivity mapping. Comparative evaluation of linear vs. nonlinear image matrix operations.

Unit 2:

Image Enhancement in the Spatial Domain: Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters.

Image Enhancement in the Frequency Domain: Introduction to the Fourier Transform and the Frequency Domain, smoothing and sharpening frequency domain

Unit 3

Image Restoration: A Model of the Image degradation/Restoration process, Noise Models, Restoration in the Presence of Noise Only—Spatial Filtering, Inverse Filtering.

Color Fundamentals: Color Models, Pseudo color Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and Sharpening, Color Segmentation, Noise in Color Images, Color Image Compression.

4. Suggested Books and References

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. New York, NY, USA: Pearson, 2018.
- [2] A. K. Jain, *Fundamentals of Digital Image Processing*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 1989.
- [3] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4th ed. Hoboken, NJ, USA: John Wiley & Sons, 2007.
- [4] S. Sridhar, *Digital Image Processing*, 2nd ed. New York, NY, USA: Oxford University Press, 2016.
- [5] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, 3rd ed. Knoxville, TN, USA: Gatesmark Publishing, 2020.

mtc-303a Big Data Analysis	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

Paradigm Shifts & Verticals: To evaluate the architectural evolution, structural traits, and core market drivers of Big Data across high-impact industry verticals.

Ecosystem Engineering: To master foundational open-source distributed frameworks, focusing on core Hadoop storage setups and multi-environment deployment vectors.

NoSQL Architecture: To analyze, compare, and implement non-relational database models, aggregate topologies, and distributed horizontal sharding mechanics.

Distributed Computation: To design and optimize complex parallel data processing engines using decoupled MapReduce scheduling, partitioning, and pipeline composition.

Perimeter Analytics: To engineer analytical architectures that span mobile business intelligence engines, crowdsourced data streams, and secure cross-firewall data structures.

2. Course Outcomes (COs)

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

CO1: Assess enterprise data infrastructure constraints by choosing the ideal distributed layout across RDBMS, Grid, and Hadoop ecosystems.

CO2: Formulate tailored analytical pipelines for specialized industry domains including real-time algorithmic trading, fraud mitigation, and healthcare informatics.

CO3: Architect scalable non-relational storage designs by configuring Key-Value, Document, and Graph NoSQL structures optimized via horizontal sharding.

CO4: Write optimized, multi-stage parallel MapReduce execution scripts integrated with custom combiners, partitioners, and chained workflows.

CO5: Build secure analytics frameworks capable of aggregating crowdsourced, mobile business intelligence, and trans-firewall data streams.

3. Course Syllabus

mtc-303a Big Data Analysis	Duration: 3 hours Max Marks: 70
---	--

Unit 1

What is big data, why big data, data, data storage and analysis, comparison with other systems, relational database management system, grid computing, volunteer computing, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies

Unit 2

Introduction to the open-source Apache Hadoop framework. Synergistic alignment between Cloud infrastructures and Big Data processing scale. Core structural mechanics of the Hadoop Distributed File System (HDFS) and Yet Another Resource Negotiator (YARN) resource scaling engines. MapReduce architecture of parallel computation. Function of Map phase, internal framework shuffling and sorting pipelines, and the final Reduce execution layer. Composing, chaining, and structuring multi-stage MapReduce pipelines for complex analytical operations.

Unit 3

Introduction to NoSQL, Limitations of ACID compliance at scale and the structural adoption of the CAP Theorem (Consistency, Availability, Partition Tolerance). Architectural taxonomy of NoSQL systems and the mechanics of schema-less storage engines. Aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schema less databases, materialized views. Performance optimization using query-time Materialized Views. Distributed scaling paradigms: Horizontal sharding methodologies, replication models, master-slave vs. peer-to-peer distribution networks, and multi-version concurrency control.

4. Suggested Books and References

- [1] T. White, Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale, 4th ed. Sebastopol, CA, USA: O'Reilly Media, 2015.
- [2] P. J. Sadalage and M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, 1st ed. Boston, MA, USA: Addison-Wesley Professional, 2012.
- [3] N. Marz and J. Warren, Big Data: Principles and Best Practices of Scalable Real-time Data Systems, 1st ed. Shelter Island, NY, USA: Manning Publications, 2015.
- [4] E. Sammer, Hadoop Operations: A Guide for Developers and Administrators, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2012.
- [5] Apache Hadoop Documentation, The Apache Software Foundation, Wakefield, MA, USA, 2026. [Online]. Available: apache.org

mtc-303b Locational Analytics with Spatial DBMS	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

Spatial Database Foundations: To master the structural mechanics of Spatial DBMS architectures, geometry data types, and spatial indexing paradigms.

Geospatial Query Optimization: To engineer high-performance spatial SQL queries using standard OGC (Open Geospatial Consortium) functions for topological relationships.

Enterprise Esri Integration: To implement seamless connections between Esri ArcGIS Pro/Enterprise and relational enterprise spatial databases.

Location Intelligence Delivery: To design sophisticated locational analytics workflows, including spatial clustering, hot-spot analysis, and site selection models.

Web GIS Deployment: To architect and publish enterprise spatial database layers as cloud-native web feature services and interactive dashboards.

2. Course Outcomes (COs)

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

CO1: Design relational database schemas that accurately store, model, and index multi-dimensional spatial geometries.

CO2: Write advanced spatial SQL scripts to calculate geographic proximities, analyze topological intersections, and filter complex spatial datasets.

CO3: Configure enterprise Esri geodatabases within corporate database engines while enforcing multi-user branch versioning rules.

CO4: Apply spatial statistical models—including hot-spot tracking and suitability analysis matrices—to resolve commercial market location problems.

CO5: Deploy high-performance web mapping services and interactive spatial analytics dashboards connected directly to live enterprise database layers.

3. Course Syllabus

mtc-303b Locational Analytics with Spatial DBMS	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Introduction to Spatial Databases, Limitations of traditional relational databases in handling geographic data. Understanding Spatial DBMS architecture. Core spatial data models: Vector geometries (Points, Linestrings, Polygons) and Raster grids. Spatial Indexing & Storage, optimization for spatial data. Mechanics of R-Tree and Grid-based spatial indexing. Understanding spatial bounding boxes, Spatial SQL Operations, spatial measurement functions (ST_Distance, ST_Area), topological relationship predicates (ST_Contains, ST_Intersects, ST_Within), and constructive geometric operations (ST_Buffer, ST_Intersection, ST_Union).

Unit 2:

Overview of Esri's location analytics ecosystem, ArcGIS Pro, ArcGIS Enterprise, and ArcGIS Online. Establishing database connections from desktop clients to relational database management systems. Converting a standard database into an Esri Enterprise Geodatabase. Understanding system tables, spatial references (EPSG coordinates), and coordinate projection systems. Implementing geodatabase topology rules to maintain data cleanliness. Understanding multi-user editing pipelines:

Unit 3:

Designing data-driven site selection models using suitability analysis techniques. Network analyst concepts: routing, service areas, and closest facility optimization using spatial data tables, spatial data mining, Spatial autocorrelation, Hot-Spot Analysis, and cluster-outlier identification. Publishing enterprise spatial database layers as web services (Map Services, Feature Services). Optimizing service performance via database views and query layers. Building operational real-time location intelligence dashboards for corporate decision-making.

4. Suggested Books and References

- [1] S. Shekhar and S. Chawla, *Spatial Databases: A Tour*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2003. (*Foundational text for Spatial DBMS mechanics*).
- [2] N. J. Obermeyer and J. K. Pinto, *Managing Geographic Information Systems*, 2nd ed. New York, NY, USA: The Guilford Press, 2008.
- [3] R. G. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2019.
- [4] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Science and Systems*, 4th ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.
- [5] *ArcGIS Enterprise Documentation & Best Practices*, Esri Inc., Redlands, CA, USA, 2026. [Online]. Available: arcgis.com

mtc-304a Software Testing	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

Testing Methodologies: To master the structural taxonomy of software defects, verification and validation frameworks, and the core software testing life cycle (STLC).

Test Case Engineering: To design optimized, comprehensive test case suites using standard black-box and white-box test design techniques.

Web & API Automation: To engineer reliable web automation suites and automated API testing layers using modern programmatic frameworks.

Non-Functional Testing: To evaluate performance metrics, load constraints, stress tolerances, and core security threat vulnerabilities of applications.

CI/CD Pipeline Integration: To integrate automated test verification infrastructure into modern DevOps continuous integration and deployment pipelines.

2. Course Outcomes (COs)

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

CO1: Build comprehensive software testing lifecycle blueprints by generating requirement traceability matrices and defect lifecycle tracking sheets.

CO2: Apply black-box and white-box mathematical coverage rules to calculate code complexity and engineer optimal test cases.

CO3: Architect scalable, clean-code web automation testing frameworks utilizing the Page Object Model design pattern.

CO4: Design automated script suites to validate REST API contracts, status codes, and structural JSON schemas.

CO5: Execute distributed performance load profiles and integrate automated testing suites directly into DevOps continuous integration pipelines.

3. Course Syllabus

mtc-304a Software Testing	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Introductory concepts: Verification & Validation Software Testing Life Cycle (STLC), STLC phases: Requirements analysis, test planning, test case development, test environment setup, test execution, and test cycle closure.

Levels of Testing: Unit Testing, Integration Testing, System Testing, Debugging, Domain Testing, Regression Testing, Stress Testing, Slice based testing.

Unit 2

Verification Testing: Verification Methods, SRS Verification, Software Design Document Verification, Code Reviews, User Documentation Verification, Software Project Audits.

Functional Testing techniques: Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique. Structural Testing: Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing.

Unit 3

Object Oriented Testing: Class Testing, GUI Testing. Agile testing methodologies, Scrum frameworks, sprint planning, and definition of done (DoD). Test-Driven Development (TDD) vs. Behavior-Driven Development (BDD). Integrating automation scripts into continuous integration (CI/CD) pipelines to execute unattended test runs on distributed container agents. Software Testing Tools, Load Runner, Win runner and Rational Testing Tools, Java Testing Tools.

4. Suggested Books and References

- [1] P. Ammann and J. Offutt, *Introduction to Software Testing*, 2nd ed. Cambridge, UK: Cambridge University Press, 2016.
- [2] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
- [3] L. Crisp and J. Gregory, *Agile Testing: A Practical Guide for Testers and Agile Teams*, 1st ed. Boston, MA, USA: Addison-Wesley Professional, 2009.
- [4] Unmesh Gundecha, *Selenium Testing Tools Cookbook*, 2nd ed. Birmingham, UK: Packt Publishing, 2015.
- [5] *OWASP Top 10 Application Security Risks*, Open Web Application Security Project (OWASP) Foundation, 2026. [Online]. Available: owasp.org

mtc-304b Natural Language Processing	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

Text Normalization Foundations: To master the structural components of natural language pipelines, focusing on text normalization, regular expressions, and minimum edit distance calculation.

Language Modeling Mechanics: To analyze and implement statistical language modeling paradigms using N-grams, smoothing methodologies, and Naive Bayes classifiers for sentiment discovery.

Sequence Labeling & Part-of-Speech: To evaluate sequence labeling pipelines by designing Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MEMMs) for part-of-speech tagsets.

Syntactic Structure Parsing: To engineer constituency and dependency parsing frameworks using dynamic programming (CKY) and probabilistic context-free grammars (PCFGs).

Computational Semantics & QA: To formulate logical meaning representations, information extraction workflows, and information retrieval (IR)-based factoid question-answering conversational systems.

2. Course Outcomes (COs)

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

CO1: Build automated text processing scripts to compute string distances, normalize unstructured corpora, and clean linguistic data.

CO2: Deploy N-gram language models and probabilistic Naive Bayes classifiers while applying advanced smoothing techniques to manage unseen vocabulary.

CO3: Implement statistical sequence-labeling models using HMMs and MEMMs to resolve parts-of-speech ambiguity in morphologically rich languages.

CO4: Design probabilistic context-free grammar constraints to parse syntactic sentences using dynamic CKY and dependency-tree graph algorithms.

CO5: Architect enterprise information extraction components to extract named entities, events, and relations while constructing conversational chatbot logic.

3. Course Syllabus

mtc-304b Natural Language Processing	Duration: 3 hours Max Marks: 70
---	--

Unit 1

The introduction of NLP. classical problems of NLP. Regular Expressions, Words, Corpora, Text Normalization, Minimum Edit Distance. N-Grams, Evaluation of Language Models, Generalization and Zeros, Smoothing. Naive Bayes Classifiers and sentiment analysis. English Word Classes, The Penn Treebank Part-of-Speech Tagset, Part-of-Speech Tagging, HMM Part-of-Speech Tagging, Maximum Entropy Markov Models, Bidirectionality, Part-of-Speech Tagging for Morphologically Rich Languages

Unit 2

Grammar: Constituency, Context-Free Grammars, Some Grammar Rules for English, Treebanks, Grammar Equivalence, and Normal Form, Lexicalized Grammars, Constituency Parsing, Ambiguity, CKY Parsing: A Dynamic Programming Approach, Partial Parsing, Statistical Constituency Parsing. Probabilistic Context-Free Grammars, Probabilistic CKY Parsing of PCFGs, Ways to Learn PCFG Rule Probabilities, Problems with PCFGs. Improving PCFGs by Splitting Non-Terminals, Probabilistic Lexicalized CFGs, Probabilistic CCG Parsing, Evaluating Parsers.

Unit 3

Dependency Relations, Dependency Formalisms, Dependency Treebanks, Transition-Based Dependency Parsing, Graph-Based Dependency Parsing, Logical Representations of Sentence- Meaning, Computational Desiderata for Representations, Model-Theoretic Semantics, First- Order Logic, Event, and State Representations, Description Logics, Information Extraction, Named Entity Recognition, Relation Extraction, Extracting Times, Extracting Events and their Times, Template Filling. Word Senses and WordNet, IR-based Factoid Question Answering, Knowledge-based Question Answering, Chatbots.

4. Suggested Books and References

- [1] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2023. (Primary textbook mapping directly to the provided core syntax, PCFG, and parsing topics).
- [2] C. D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing, 1st ed. Cambridge, MA, USA: MIT Press, 1999.
- [3] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2009.
- [4] P. Wagh, Natural Language Processing: Foundations, Models, and Applications, 1st ed. Saarbrücken, Germany: LAP Lambert Academic Publishing, 2024.
- [5] S. Ghosh and D. Gunning, Natural Language Processing Fundamentals: Build Intelligent Applications That Can Interpret the Human Language to Deliver Impactful Results, 1st ed. Birmingham, UK: Packt Publishing, 2019.

mtc-401 Advanced Python for Data Science	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

Python Engineering Foundations: To master advanced core Python constructs, functional programming patterns, and object-oriented paradigms tailored for data management.

Vectorized Data Processing: To engineer high-performance numerical computing modules using multi-dimensional array vectorization and matrix manipulation scripts.

Structured Data Wrangling: To perform production-grade data cleaning, structural merging, group aggregations, and exploratory timeseries analysis using data frames.

Statistical Visualization: To design publication-ready, data-driven visualizations using programmatic canvas configurations and high-level statistical chart mapping.

Predictive Pipeline Synthesis: To build, evaluate, and fine-tune scalable machine learning pipelines for supervised classification, continuous regression, and unsupervised clustering.

2. Course Outcomes (COs)

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

CO1: Write optimized, modular Python scripts using functional programming syntax, list comprehensions, and robust custom exception pipelines.

CO2: Compute vectorized numerical calculations and high-speed linear algebra operations by designing multi-dimensional NumPy array masks.

CO3: Perform complete data cleaning and structured wrangling transformations on messy business datasets using Pandas split-apply-combine methods.

CO4: Generate publication-grade statistical charts and exploratory visualization subplots to analyze complex data patterns.

CO5: Deploy production-ready machine learning pipelines to execute classification, regression, and clustering tasks verified by multi-metric statistical evaluation.

3. Course Syllabus

mtc-401 Advanced Python for Data Science	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Introduction to the data science lifecycle, development environments, and scripting execution patterns. Core data type systems: Lists, Tuples, Dictionaries, and Sets. Functional Control Flows, Advanced control structures, list comprehensions, dictionary transformations, and generator expressions. Designing functions, variable scopes, lambda constructs, Multi-dimensional array structures (ndarray).

Unit 2

Data Frame Architectures, Introduction to structural abstractions, Series components and DataFrames. Indexing mechanisms, data alignment properties, and importing data from distinct formats (CSV, Excel, SQL relational engines). Advanced data cleaning protocols: handling missing values, duplicate extraction, and string transformation pipelines. Structural joins, column mapping, concatenation matrices, and hierarchical index formatting. Temporal analysis frameworks: date parsing, period windows, rolling statistics, and resampling frequencies. Exploratory data analysis visualization, histograms, correlation scatter plots, and statistical maps.

Unit 3

Supervised Learning Engines, architecture of predictive classifiers and regression models. Neighborhood distance mapping (k-NN) and structural Naive Bayes probabilistics. Non-linear decision boundary design via Support Vector Machines (SVM). Decision Trees and high-throughput Random Forest models.

Unsupervised Clustering & Validation: Centroid-based space partitioning (k-means topology) and agglomerative hierarchical clustering metrics. Validation mechanics: train-test data splitting, stratified k-fold cross-validation frameworks, and grid search hyperparameter tuning. Model accuracy telemetry using confusion matrices, precision, recall, F1-scores, and ROC-AUC curve vectors.

4. Suggested Books and References

- [1] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [2] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [3] J. VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [4] A. Mueller and S. Guido, Introduction to Machine Learning with Python: A Guide for Data Scientists, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2016.
- [5] Python Software Foundation & Scikit-Learn Ecosystem Reference, Python Software Foundation, Wilmington, DE, USA, 2026. [Online]. Available: scikit-learn.org

mtc-403a Business Forecasting and Predictive Analytics	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the theoretical mathematics of generalized linear models and time-to-event survival analysis within corporate data pipelines.
- Understand the structural identification, tuning, and optimization of multi-component stochastic time-series forecasting networks (SARIMAX/VAR).
- Apply high-performance non-linear ensemble machine learning algorithms (XGBoost, Random Forests) to solve highly unbalanced business classification problems.
- Implement neural network sequence configurations to predict long-horizon corporate demand and financial market variables.
- Evaluate predictive machine learning architectures using model diagnostics and explainable AI frameworks (SHAP/LIME) to justify operational automated decisions to executive stakeholders.

2. Course Outcomes (COs)

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

CO-1: Construct unified feature preprocessing pipelines to ingest categorical matrices and fit regularized elastic net or Cox proportional hazard models.

CO-2: Design and troubleshoot multi-seasonal SARIMAX forecasting pipelines, evaluating stationarity properties and residual error constraints.

CO-3: Formulate deep sequence recurrent networks (LSTMs) to parse multi-variable historical inputs and output multi-step trend projections.

CO-4: Model automated tree-based boosting architectures (XGBoost/LightGBM) by setting custom objective loss structures and executing stratified validation loops.

CO-5: Assemble a complete explainable AI diagnostic portfolio using SHAP parameters to isolate feature weights and optimize classification thresholds under strict enterprise cost limitations.

3. Course Syllabus

mtc-403a Business Forecasting and Predictive Analytics	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Introduction to Business forecasting, Data Driven Decision Making and Essentials of Predictive Analytics, Types of Forecasting: Qualitative Approaches and Quantitative Approaches, Components of a Time Series and Measures of Forecast Accuracy, Human Judgement in Time Series Analysis, Regression Models, Survival Analysis, Generalized Linear Models (GLMs), maximum likelihood estimation processes. Multi-class classification systems using Multinomial Logistic Regression; Survival Analysis, Churn Modelling, Modelling time-to-event business data, estimate customer lifetime value and employee attrition rates. Regularization and Feature Selection, tuning hyperparameters via elastic net regularization paths.

Unit 2

Advanced Time-Series Forecasting, Deep Sequence Models, Stochastic Longitudinal Modeling, Foundations of classical time-series decompositions (trend, cyclical, seasonal components); evaluating structural stationary properties using Augmented Dickey-Fuller (ADF) tests. Dynamic Exogenous Scaling, Incorporating external corporate drivers into time-series frameworks using ARIMAX and Vector Autoregression (VAR) systems to evaluate cross-variable economic interactions. State-Space Models and Structural Forecasting, exponential smoothing state-space models (ETS);

Unit 3

Ensemble Machine Learning Workflows, Tree-Based Non-Linear Architectures, building high-capacity decision models using Bagging (Bootstrap Aggregation) and Random Forests; interpreting out-of-bag (OOB) errors and feature importance distributions. Gradient Boosting Frameworks: Sequential optimization models; algorithm engineering via Adaptive Boosting (AdaBoost), Gradient Boosted Decision Trees (GBDT), and high-performance framework implementations (XGBoost / LightGBM). Explainable AI (XAI) in Business.

4. Suggested Books and References

- [1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd ed. New York, NY, USA: Springer, 2021.
- [2] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.
- [3] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [4] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021.
- [5] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 2nd ed. Munich, Germany: Leanpub, 2022.

mtc-403b	Duration: 3 hours
Business Intelligence using Locational Analytics	Max Marks: 70

1. Course Objectives

Foundations of Geo-Intelligence: To understand the convergence of legacy Business Intelligence (BI) frameworks with modern spatial data models and location-based telemetry.

Spatial ETL & Data Warehousing: To design spatial extract-transform-load (ETL) data pipelines that clean, enrich, and warehouse transactional business logs with geographic attributes.

Geodemographic Analytics: To master advanced spatial demographic modeling, including consumer drive-time trade areas, catchment zones, and customer density metrics.

Spatial Market Mining: To apply spatial statistics, cluster mapping, and predictive site-suitability models to optimize corporate physical footprint expansion.

Interactive Geo-Dashboards: To architect production-grade, real-time spatial dashboards and map-based KPI tracking interfaces for executive enterprise delivery.

2. Course Outcomes (COs)

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

CO1: Build automated spatial data pipelines to standardize, geocode, and enrich traditional transactional databases with geographic parameters.

CO2: Compute precise enterprise trade areas, driving catchments, and market gravity projections by executing network routing matrix scripts.

CO3: Implement geodemographic classification and spatial allocation models to estimate localized store cannibalization and market share boundaries.

CO4: Design multi-criteria spatial suitability models using weighted overlays to resolve location problems such as corporate site selection.

CO5: Architect enterprise-grade business intelligence dashboards that map active performance metrics, KPIs, and spatial hot-spots for corporate governance.

3. Course Syllabus

mtc-403b	Duration: 3 hours
Business Intelligence using Locational Analytics	Max Marks: 70

Unit 1

Business Intelligence Evolution, Spatial Data Infrastructure, & ETL, Defining Location Intelligence (LI) and its role in enterprise decision-making. Core traits of spatial data models inside enterprise architectures, Integrating demographic data, human mobility streams (foot traffic), point-of-interest (POI) registers, and localized vehicle traffic volumes. Spatial ETL & Warehousing Operations: Extending classic Data Warehousing to support spatial types.

Unit 2

Trade Area Modeling (Catchment Analysis): Defining customer catchment spaces. Creating simple proximity rings (Euclidean distance buffers) vs. complex network-based buffers. Calculating definitive travel-time and drive-distance service polygons using routing engines and historical congestion datasets.

Geodemographic Segmentation: Applying spatial classification algorithms to group market populations.

Network & Routing Optimization: Logistics engineering in enterprise BI. Solving the closest facility problem for field service routing. Multi-stop delivery routing optimization, supply chain service boundary mapping, and distribution hub allocation.

Unit 3

Executing spatial data mining over sales performance records: Spatial Autocorrelation (identifying clustered market pockets), Hot-Spot Analysis (Getis-Ord G_i^*) for locating micro-markets with high sales velocity, and spatial outlier detection to track underperforming stores. Multi-Criteria Site Suitability Analysis: Designing data-driven corporate site-selection models. Enterprise Mapping & Geo-Dashboards: Designing executive BI dashboards that integrate dynamic interactive mapping canvases. Mapping classic business metrics and KPIs (Year-over-Year growth, customer acquisition costs, churn) onto regional and neighborhood choropleth spaces.

4. Suggested Books and References

[1] C. Harder, *The Business Benefits of GIS: An ROI Approach*, 1st ed. Redlands, CA, USA: Esri Press, 2010.

[2] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2016.

[3] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, 1st ed. New York, NY, USA: McGraw-Hill Education, 2013.

[4] E. Turban, R. Sharda, and D. Delen, *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*, 4th ed. Boston, MA, USA: Pearson, 2017.

[5] *Gartner Magic Quadrant for Analytics and Business Intelligence Platforms*, Gartner Inc., Stamford, CT, USA, 2026. [Online]. Available: gartner.com

[6] J. Pick, *Geo-Business: GIS in the Digital Organization*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2008.

mtc-404a Object Oriented Software Engineering	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

To master the engineering life cycle transitions from structured requirements analysis to scalable object-oriented design paradigms.

To express system mechanics visually by building comprehensive structural, behavioral, and architectural UML blueprints.

To evaluate, select, and implement structural, creational, and behavioral design patterns to solve common software design limitations.

To design, build, and execute robust validation strategies, including unit, integration, and system-level object-oriented testing frameworks.

To manage multi-tenant engineering workflows using Agile frameworks, automated version controls, and continuous deployment tracking.

2. Course Outcomes (COs)

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

CO1: Formulate detailed requirements specifications by writing formal use cases and establishing software domain analysis frameworks.

CO2: Model complex software properties by designing high-fidelity structural and behavioral UML diagrams.

CO3: Architect modular software systems that conform to strict SOLID design patterns and multi-tier architectural standards.

CO4: Write object-relational mapping logic and production-ready source code that translates UML maps directly into functional programs.

CO5: Build automated test case suites and deploy code adjustments within automated continuous integration pipelines.

3. Course Syllabus

mtc-404a Business Intelligence and OLAP Cubes	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Object-Oriented Analysis (OOA): Requirements elicitation and domain engineering. Use-case driven approaches: Identifying actors, establishing use-case boundaries, writing formal scenarios, and capturing extensions/preconditions. Translating system requirements into high-level analysis models. Introduction to the Unified Modeling Language (UML) grammar. Constructing Class Diagrams: Classes, operations, attributes, and visibility constraints. Modeling structural associations: Generalization, aggregation, composition, and multi-faceted multiplicity rules.

Unit 2

System Design Principles: Transitioning from analysis to architecture. Object-oriented design paradigms: Cohesion, coupling, encapsulation optimization, and the SOLID principles (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion). Designing multi-tier system layers: Model-View-Controller (MVC), Three-Tier Client-Server layouts, and Microservices foundations. Introduction to Gang of Four (GoF) design patterns: Creational (Singleton, Factory Method, Abstract Factory), Structural (Adapter, Composite, Facade, Proxy), and Behavioral (Observer, Strategy, Command, State).

Unit 3

Database mapping: Translating object schemas into Relational Database Management Systems (RDBMS) via Object-Relational Mapping (ORM) design patterns and handling inheritance hierarchies in tables. Object-Oriented Testing Frameworks: Unique verification challenges in OO systems (polymorphism, inheritance, and encapsulation side-effects). White-box and black-box testing adaptations. Designing automated Unit Testing suites, mock object injection, Class Testing, and Integration Testing strategies (Top-down, Bottom-up, and Sandwich approach). System and regression testing tools.

4. Suggested Books and References

- [1] B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering Using UML, Patterns, and Java*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2009. (*Primary text for baseline structural modeling and engineering concepts*).
- [2] R. S. Pressman and B. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill Education, 2019.
- [3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. Boston, MA, USA: Addison-Wesley Professional, 1994.
- [4] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, 1st ed. Reading, MA, USA: Addison-Wesley Professional, 1992.
- [5] *IEEE Transactions on Software Engineering (TSE)*, IEEE Computer Society, Washington, DC, USA, 2026. [Online]. Available: computer.org

mtc-404b Deep Learning and Generative AI	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

- **Understand** the structural, mathematical, and algorithmic foundations of deep feedforward neural networks and vector-based backpropagation loops.
- **Master** the selection, layout tuning, and performance optimization of multi-layer weight regularizations, normalization techniques, and gradient drop mitigations.
- **Apply** convolutional neural networks and advanced spatial architectures to execute high-accuracy computer vision classification and object location pipelines.
- **Implement** gated recurrent sequence processing structures (LSTMs/GRUs) to model complex historical multi-variable data and language sequences.
- **Evaluate** multi-head self-attention transformers and generative deep learning frameworks (GANs, VAEs, Diffusion models) to engineer state-of-the-art intelligent systems.

2. Course Outcomes (COs)

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

- **CO-1:** Construct deep feedforward tensor models utilizing customizable weight initialization matrix parameters, Adam optimizers, and dropout layers.
- **CO-2:** Design and optimize deep Convolutional Neural Networks (CNNs) incorporating residual skip paths to process multidimensional spatial image arrays.
- **CO-3:** Formulate gated recurrent network structures (LSTMs/GRUs) to analyze chronological dependencies and process sequence-to-sequence language transformations.
- **CO-4:** Model multi-head self-attention transformer building blocks to execute semantic feature token alignments.
- **CO-5:** Assemble deep generative systems using adversary or latent space autoencoder structures to synthesize unique multi-variate data matrices.

3. Course Syllabus

mtc-404b Deep Learning and Generative AI	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Foundations of Deep Networks, Deep Feedforward Networks, Linear threshold units, multi-layer perceptron (MLP) mapping. Regularization & Training Optimization, Diagnosing vanishing/exploding gradients; Layer Normalization, Dropout constraints, and Weight Decay (L_2 regularization). Convolutional Neural Networks (CNNs): Spatial invariance principles; convolution operations (filters, stride, padding, dilations), pooling mechanics (Max/Average pooling), and fully connected output mappings.

Unit 2

Object Detection & Segmentation Basics: Introduction to anchor boxes, bounding box regressions (YOLO/SSD), and pixel-level spatial semantic segmentation layouts (U-Net architectures). Recurrent Neural Networks (RNNs) & Sequence Modeling: Processing temporal text loops; unfolding computational structures over time, exploding/vanishing backpropagation-through-time (BPTT) challenges, and bi-directional cells. Gated Sequence Architectures, Overcoming long-term dependencies; internal gating mechanisms of Long Short-Term Memory (LSTM) networks (forget, input, and output gates) and Gated Recurrent Units (GRUs).

Unit 3

Attention Networks, Transformers, Generative Systems Sequence-to-sequence mapping bottlenecks; introduction to cognitive attention models. The Transformer Architecture Blueprint, Deconstructing the complete encoder-decoder layout; Multi-Head Attention mechanisms, positional encoding matrices, feedforward sub-networks, and residual layer connections. Foundational Model Overviews, Encoder-only models (BERT), Decoder-only autoregressive tracks (GPT models), and Encoder-Decoder paradigms (T5). Generative Adversarial Networks (GANs): Zero-sum game modeling; architectural coordination between a Discriminator network and a Generator network.

4. Suggested Books and References

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [2] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.
- [3] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Hoboken, NJ, USA: Pearson, 2023.
- [4] C. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*, 1st ed. Cham, Switzerland: Springer, 2023.
- [5] PyTorch Engineering Team. *PyTorch Documentation: Core Tensors, Neural Network Layer Modules, and Optimizers*. [Online]. Available: <https://pytorch.org>
- [6] TensorFlow Team. *TensorFlow Core Keras API Reference Manual for Layer Construction and Model Compilation*. [Online]. Available: <https://tensorflow.org>