

MAHARSHI DAYANAND SARASWATI UNIVERSITY

AJMER



Syllabus

**Scheme of Examination and
Courses of Study**

Faculty of Science

M.Sc. (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

M.Sc. (Computer Science)

Program Code:

Program Faculty: Computer Science

Program Name: M.Sc. (Computer Science)

1. Introduction to the Program

The Master of Science in Computer Science (M.Sc. CS) is an advanced, two-year post-graduate program structured to blend foundational computer science theory with state-of-the-art technological applications. Operating under the Faculty of Science, this curriculum is systematically split across four semesters to provide deep technical proficiency.

The program bridges fundamental disciplines—such as computer systems architecture, open-source Linux technologies, and advanced data structures—with modern engineering domains like cloud networking, full-stack web development, and mobile computing. To meet the demands of the evolving tech sector, the curriculum includes dedicated elective tracks in Data Analytics, Machine Learning, Deep Learning, and Generative AI. Students balance rigorous classroom instructions with extensive advanced laboratory sessions and a mandatory end-semester research dissertation, transforming them into adaptive systems engineers, data architects, and technical researchers.

2. Objectives of the Program

The primary objectives of the M.Sc. Computer Science program are to:

Establish Computational Core: Build strong theoretical foundations in computer organization, operating systems, and algorithmic analysis.

Cultivate Software Engineering: Develop full-stack software development skills across multi-tier ecosystems using Python, Java, PHP, and JavaScript.

Drive Next-Gen Intelligence: Train students in advanced data paradigms, covering data warehousing, predictive business analytics, and neural network architectures.

Master Enterprise Infrastructure: Provide practical command over classless routing architectures, enterprise switching, and distributed cloud computing systems.

Promote Applied Research: Engage students in independent scientific inquiry through software prototyping, interface design (UI/UX), and dissertation research.

3. Employment and Entrepreneurial Scope

Employment Scope

Graduates of this program are qualified for specialized technical roles within corporate IT sectors, research labs, public institutions, and technology firms:

Full-Stack & Web Developer: Engineering highly reactive user interfaces and server-side back-end engines using PHP, Java, and JavaScript (UI/UX).

Mobile Application Engineer: Designing native, high-performance Android applications using modern SDK architectures and background execution threads.

Data Scientist / ML Engineer: Deploying supervised/unsupervised machine learning models, ETL data pipelines, and deep neural network architectures.

Cloud & Network Administrator: Configuring enterprise subnetting, VLAN topologies, routing architectures, and scalable cloud network spaces.

Business Intelligence Analyst: Constructing multi-dimensional data cubes, managing OLAP servers, and designing analytical enterprise dashboards.

Entrepreneurial Scope

The technical depth of the MDSU M.Sc. CS syllabus lays down an ideal path for tech-driven business ventures:

SaaS Product Outfits: Building standalone software platforms or mobile apps targeting niche enterprise or consumer workflows.

Analytics & AI Consultancies: Provisioning client-centric data mining, business intelligence setups, and data warehousing migrations.

Independent Web Development Agencies: Launching boutique digital agencies specializing in corporate infrastructure deployment, UI/UX optimization, and web frameworks.

Open-Source System Integration: Building businesses around customized Linux infrastructure configurations, hardware-software deployments, and network optimizations.

4. Learning Outcomes of the Program (PO)

Upon successful completion of the M.Sc. Computer Science program, graduates will demonstrate the following Program Outcomes:

(PO-1) Advanced Technical Knowledge: Apply foundational computing concepts across computer systems architecture, database management (DBMS), and open-source operating systems.

(PO-2) Problem Analysis & Algorithmic Optimization: Analyze complex computation problems, evaluating the space and time complexities of recursive and non-recursive algorithms to optimize processing limits.

(PO-3) Full-Stack Design & Development: Architect scalable, multi-tier web and enterprise systems utilizing structured design principles, clean data schemas, and object-oriented paradigms.

(PO-4) Mobile Ecosystem Engineering: Build secure, data-driven mobile computing solutions that interact with external web services and manage platform background states seamlessly.

(PO-5) Modern Tool & Infrastructure Usage: Configure and manipulate industry-standard toolchains including the Android SDK, Cisco switching environments, Git/Linux environments, and Figma layout grids.

(PO-6) Intelligent Systems & Analytics Modeling: Extract knowledge patterns from high-density datasets by constructing data mining pipelines, machine learning matrices, and generative prompt spaces.

(PO-7) Research Inquiry & Problem Scoping: Conduct systemic literature reviews, design test scripts, isolate driver variables, and execute an independent scientific dissertation.

(PO-8) Communication & Documentation: Draft precise Software Requirements Specifications (SRS), build clear project wireframes, and deliver professional technical defenses of complex software assets.

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	mcs-101 DCC – 01	Computer System Architecture	3	1	0	4	30	70	100	All Courses are compulsory Courses
2	mcs-102 DCC – 02	Programming with Python	3	1	0	4	30	70	100	
3	mcs-103 DCC – 03	Electronic Data Processing and DBMS	3	1	0	4	30	70	100	
4	mcs-104 DCC – 04	Open-Source Technology and Linux	3	1	0	4	30	70	100	
5	mcs-105 DCC – 05	Lab - Python and Database Management Systems	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	mcs-201 DCC – 06	Web Development	3	1	0	4	30	70	100	
7	mcs-202 DCC – 07	Enterprise Networking and Cloud Networks	3	1	0	4	30	70	100	
8	mcs-203 DCC – 08	Programming in Java	3	1	0	4	30	70	100	
9	mcs-204a DSE/GEC – 01A	Software Engineering	3	1	0	4	30	70	100	Student may opt any one
	mcs-204b DSE/GEC – 01B	Data Structures and Algorithms	3	1	0	4	30	70	100	
10	mcs-205 DCC – 09	Lab-Web Development and Java	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	mcs-301 DCC – 10	Mobile Application Development	3	1	0	4	30	70	100	
12	mcs-302 DCC – 11	Data Warehousing & Data Mining	3	1	0	4	30	70	100	
13	mcs-303a DSE – 02A	Business Analytics	3	1	0	4	30	70	100	Student may opt any one
	mcs-303b DSE – 02B	Artificial Intelligence and Heuristics								
14	mcs-304a DSE – 03A	Data Analytics and Insights	3	1	0	4	30	70	100	Student may opt any one
	mcs-304b DSE – 03B	Machine Learning								
15	mcs-305 DCC – 12	Lab-Mobile Application Development and Data Mining	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	mcs-401 DCC – 13	JavaScript with UI /UX User Centric Design Principles	3	1	0	4	30	70	100	
17	mcs-402 DCC – 14	Dissertation	3	1	0	4	30	70	100	
18	mcs-403a DSE – 04A	Predictive Business Analytics	3	1	0	4	30 30	70 70	100	Student may opt any one
	mcs-403b DSE – 04B	Generative AI & Prompt Engineering								
19	mcs-404a DSE – 05A	Business Intelligence and OLAP	3	1	0	4	30 30	70 70	100	Student may opt any one
	mcs-404b DSE – 05B	Deep Learning & Neural Network Architectures								
20	mcs-405 DCC – 15	Lab Java Script with UI/UX User Centric Design Principles	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 A 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 B 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 C 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

(M. Sc. Computer Science)

Reg. 17 (a)

The M. Sc. 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. Sc. Computer Science : 50% marks in any graduation scheme.
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.

mcs-101 Computer System Architecture	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the fundamental building blocks of digital logic circuits, data representations, and the syntax of Register Transfer Language (RTL) for hardware operations.
- Understand the structural organization, execution behaviors, microprogrammed control logic, and instruction cycles of a baseline computer system.
- Analyze the internal architectures of the Central Processing Unit (CPU), contrasting register structures, addressing modes, and RISC/CISC processing philosophies.
- Examine high-performance hardware strategies including pipeline execution matrices, vector architectures, and parallel processing configurations.
- Evaluate memory hierarchy optimization methods (cache, associative, and virtual memory systems) alongside multi-processor coordination and bus arbitration protocols.

2. Course Outcomes (COs)

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

CO-1: Design optimized combinational and sequential digital logic circuits and perform hardware-level execution of arithmetic, logical, and shifting microoperations.

CO-2: Map out the exact step-by-step lifecycle of an instruction through fetch, decode, and execute cycles, handling internal register changes and interrupt loops.

CO-3: Assemble lower-level machine and assembly routines utilizing explicit addressing loops, subroutine linkages, and input/output isolated programming.

CO-4: Evaluate pipeline processing performance metrics to eliminate data, control, and structural hazards within arithmetic and instruction pipelines.

CO-5: Structuring high-efficiency memory and peripheral systems by implementing Direct Memory Access (DMA), cache mapping, virtual spaces, and cache-coherence strategies across multicore structures.

3. Course Syllabus

mcs-101 Computer System Architecture	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Digital Logic Circuits and Components: Digital Computers, Logic Gates, Boolean Algebra, Map Simplifications, Combinational Circuits, Flip-Flops, Sequential Circuits, Integrated Circuits, Decoders, Multiplexers, Registers and Counters, Memory Unit. Data Representation: Data Types, Number Systems and Conversion, Complements, Fixed Point Representation, Floating Point Representation, Error Detection Codes, Computer Arithmetic - Addition, Subtraction, Multiplication and Division Algorithms. Register Transfer and Microoperations: Register Transfer Language, Bus and Memory Transfers, Arithmetic, Logic and Shift Microoperations.

Unit 2

Basic Computer Organization and Design: Stored Program Organization and Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output, Interrupt. Programming the Basic Computer: Machine Language, Assembly Language, Assembler, Program Loops, Subroutines, Input-Output Programming. Microprogrammed Control: Control Memory, Address Sequencing, Design of Control Unit. Central Processing Unit: General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, RISC Computer, CISC Computer.

Unit 3

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, Vector Processing Array Processors. Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA, Serial Communication. Memory Hierarchy: Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware. Multiprocessors: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor Arbitration, Inter-processor Communication and Synchronization, Cache Coherence, Multicore Processors

4. Suggested Books and References

- [1] M. M. Mano, *Computer System Architecture*, 3rd ed. Englewood Cliffs, NJ, USA: Pearson Education, 1993.
- [2] W. Stallings, *Computer Organization and Architecture: Designing for Performance*, 11th ed. Hoboken, NJ, USA: Pearson Education, 2019..
- [3] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2020.
- [4] J. P. Hayes, *Computer Architecture and Organization*, 3rd ed. New York, NY, USA: McGraw-Hill Education, 1998.
- [5] Intel Corporation, *Intel 64 and IA-32 Architectures Software Developer's Manual*. [Online]. Available: intel.com.
- [6] IEEE Computer Society. *IEEE Transactions on Computers*. [Online]. Available: computer.org.

mcs-102 Programming with Python	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the core philosophy, execution mechanics, variable scoping rules, and dynamic typing architecture of the Python programming runtime environment.
- Understand foundational algorithmic controls by building structured logic using conditional branching matrices, optimization loops, and nested control paths.
- Evaluate the structural and performance differences between mutable and immutable memory entities, including sequential strings, matrices, arrays, and customized memory blocks.
- Develop intermediate to advanced collection processing patterns using native lists, immutable tuples, unique sets, and high-performance dictionary hash maps.
- Implement professional software configurations using variable-length functional frameworks, multi-file modular distribution channels, structured exception safety nets, and local file storage streams.

2. Course Outcomes (COs)

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

CO-1: Construct syntax-compliant Python scripts utilizing dynamic type bindings, complex operator hierarchies, logical structures, and standard low-level array manipulation methodologies.

CO-2: Organize text processing streams by executing string cleaning algorithms, escaping patterns, slicing transformations, and native token validation functions.

CO-3: Formulate complex data processing matrices by deploying optimal collections, manipulating multi-tier lists/tuples, enforcing set uniqueness, and engineering high-efficiency dictionary mappings.

CO-4: Design modular and scalable source code structures by engineering parameterized functions, variable-length argument trackers (*args/**kwargs), lambda configurations, and clean namespace execution layouts.

CO-5: Deploy fault-tolerant applications by constructing comprehensive exception handling blocks (try/except/finally) and orchestrating persistent data preservation pipelines via file system I/O streams.

3. Course Syllabus

mcs-102 Programming with Python	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Introduction: History, Versions, Features, Advantages, Application areas. IDLE, Editors, Keywords, Identifiers, Indents, Input Output Basic Syntax, Variable, Dynamic Typing, Data Types (Mutable and Immutable), Built-in Conversion Methods.

Operator: Arithmetic, Comparison, Logical, Identity, Membership. **Control Statements:** Conditional (If , elif, Nested if-else), Looping (While, For, Nested loops), Break, Continue, Pass. **Array:** Introduction, Creation, Traverse, Insertion, Deletion, Search, Update. **String:** Introduction, Types, Escape Sequences, Formatting, Operators, Built-in Methods (Capitalize, Upper, Lower, Title, Find, Count, Isalpha, Isdigit, Islower, Isupper), Basic Operations (Accessing, Updating, Concatenation)

Unit 2

List & Tuple: Introduction, Accessing, Operators, Built-in Methods (Len, Max, Min, Append, Insert, Remove, Pop, Reverse, Sort, List), Basic Operations (Updating, Delete, Concatenation, Indexing, Slicing). **Set:** Introduction, Accessing, Built-in Methods (Add, Update, Clear, Copy, Discard, Remove), Operations (Union, Intersection, Difference). **Dictionary:** (Single Dimensional) Introduction, Accessing, Updating, Deleting, Viewing values in dictionaries, Built-in Methods (Len, Max, Min, Pop, Clear, Items, Keys, Values, Update).

Unit 3

Function: Defining, Calling, Function Arguments (Required, Keyword, Default, Variable Length) Anonymous Functions, Global and Local Variables. **Modules:** Introduction, Importing Module, Built-in Modules (Math, Statistics, Random). **Package:** Creating, Installing, Importing Modules from the Package. **Errors & Exception:** Error Types, Exception Handling - Introduction, Try, Except, Else, Finally. **File Input-Output:** Opening and Closing files, Reading and Writing files.

4. Suggested Books and References

[1] E. Matthes, *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, 3rd ed. San Francisco, CA, USA: No Starch Press, 2023.

[2] M. Lutz, *Learning Python: Powerful Object-Oriented Programming*, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2013.

[3] J. R. Hubbard, *Schaum's Outline of Programming with Python*, 1st ed. New York, NY, USA: McGraw-Hill Education, 2018.

[4] L. Ramalho, *Fluent Python: Clear, Concise, and Effective Programming*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

[5] Python Software Foundation. *The Python Standard Library Documentation*. [Online]. Available: <https://python.org>

[6] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

mcs-103: Electronic Data Processing and DBMS	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the fundamental differences between file-processing systems and modern database management systems, highlighting the core roles of data abstraction, languages (DDL/DML), and database administration.
- Understand the mathematical foundations of the relational model, including relational algebra operations, integrity constraints, and the primary characteristics of transaction management (ACID properties).
- Develop strong conceptual and logical database design skills using Advanced Entity-Relationship (ER) modeling and normalization theory up to the Third Normal Form (3NF).
- Acquire practical, hands-on proficiency in desktop relational database design using Microsoft Access, emphasizing table creation, input validation, and data integrity enforcement.
- Apply relational query mechanisms in practical environments using both Microsoft Access Query by Example (QBE) visual grids and raw structural Access SQL commands.

2. Course Outcomes (COs)

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

CO-1: Analyze database system architectures, contrasting physical and logical data independence, and distinguish between hierarchical, network, relational, object-oriented, and NoSQL data models.

CO-2: Apply relational algebra operations and evaluate integrity constraints (domain, entity, and referential) to solve data manipulation problems and manage foreign key cascade behaviors.

CO-3: Design conceptual data models using advanced ER diagrams and systematically map them into structural relational schemas while identifying keys (super, candidate, primary).

CO-4: Demonstrate schema refinement by analyzing functional dependencies and normalizing relational structures to eliminate insertion, deletion, and modification anomalies up to 3NF.

CO-5: Build fully operational desktop database applications in MS Access by configuring validation rules, writing explicit SQL queries, and engineering functional forms, subforms, and analytical reports.

3. Course Syllabus

mcs-103: Electronic Data Processing and DBMS	Duration: 3 hours Max Marks: 70
---	--

Unit 1

File-processing systems vs. modern DBMS, objectives of database systems, data abstraction, data definition language, data manipulation language, database administrator. Database design stages, data independence, database system architecture. Overview of relational, object-oriented, and NoSQL data models.

Relational Model Constraints: Domain constraints, entity integrity, referential integrity, operational constraints, and foreign key cascade actions. Relational Algebra: Fundamental operations (Select, Project, Union, Set Difference, Cartesian Product) and joined operations (Inner, Outer, Natural joins). Transaction Management: ACID properties (Atomicity, Consistency, Isolation, Durability) and basic concurrency challenges.

Unit 2

Conceptual Modeling: Entity-Relationship (ER) modeling, identifying entities, attributes, relationships, tuples, domains, keys, super and candidate key, cardinalities, and mapping ER diagrams into relational schemas.

Functional Dependencies, Normalization Theory: Data anomalies (insertion, deletion, modification), first normal form (1NF), second normal form (2NF), and third normal form (3NF).

Unit 3

MS-Access: Create a Table in MS Access -Data Types, Field Properties, Fields names, types, properties, default values, format, caption, validation rules. Table Design & Data Integrity: Defining explicit data types, fields properties, validation rules, input masks, and establishing relationships via the Relationship Window.

Query Design & SQL View: Creating Select, Parameter, and Totals (Aggregate) queries using the Query by Example (QBE) grid, and writing raw Access SQL statements.

Introduction to Forms, Types of Basic Forms: Columnar, Tabular, Datasheet, Main/ Sub-forms. Introduction to Reports, Types of Basic Reports

4. Suggested Books and References

[1] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. New York, NY, USA: McGraw-Hill Education, 2019.

[2] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Hoboken, NJ, USA: Pearson Education, 2016.

[3] L. A. Ulrich and K. Cook, *Access 2019 Bible*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2018.

[4] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. New York, NY, USA: McGraw-Hill Education, 2003.

[5] C. Coronel and S. Morris, *Database Systems: Design, Implementation, & Management*, 13th ed. Boston, MA, USA: Cengage Learning, 2018.

[6] C. J. Date, *An Introduction to Database Systems*, 8th ed. Boston, MA, USA: Pearson Education, 2004.

mcs-104: Open-Source Technology and Linux	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the core principles, architecture, and resource management mechanics of modern enterprise operating systems.
- Analyze foundational OS mechanisms including process synchronization, CPU scheduling, deadlock management, and memory allocation.
- Examine the concepts of virtualization, distributed processing, and the philosophy behind free and open-source operating systems (FOSS).
- Develop a robust understanding of the Linux architecture, its core utilities, and standard file-system navigation mechanics.
- Understand shell scripting techniques to automate administrative tasks, handle inputs/outputs, and manage system processes efficiently.

2. Course Outcomes (COs)

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

CO-1: Differentiate between various classes of operating systems (distributed, real-time, multi-processing) and map their structural architectures to specific organizational hardware constraints.

CO-2: Evaluate CPU scheduling algorithms and synthesize programmatic solutions for core system bottlenecks like deadlocks and thrashing.

CO-3: Design and critique memory mapping frameworks using contiguous allocation, segmentation, paging, and demand-paging schemas on Intel 32- and 64-bit architectures.

CO-4: Utilize the Linux command-line interface efficiently, executing structural utilities, standard input/output redirections, pipes, and filters.

3. Course Syllabus

mcs-104: Open-Source Technology and Linux	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Introduction to Operating Systems, resource allocator and related functions, classes of OS, batch processing, multi-processing, time sharing, distributed, real time systems. Computer-System Organization, Computer-System Architecture, Operating System Operations, Resource Management, Security and Protection, Virtualization, Distributed Systems, Free and Open-Source Operating Systems. Operating-System Services, User and Operating-System Interface, System Calls, System Services, Linkers and Loaders, Why Applications Are Operating-System Specific.

Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, IPC in Shared-Memory Systems, IPC in Message-Passing Systems, threads, process scheduling, fundamental of scheduling, scheduling criteria, long medium short term scheduling, scheduling algorithms. Dead lock, characterization, methods for handling dead locks, prevention, avoidance, thrashing

Unit 2

Management of Main Memory, Logical versus physical address, swapping, contiguous allocation, segmentation, paging, segmentation with paging, kernel memory allocation, page replacement algorithm, virtual memory, virtual memory with paging, demand paging, Structure of the Page Table, Swapping, Memory management in Intel 32- and 64-bit Architectures

Unit 3

Introduction to Linux Utilities: history, architecture, and features of Linux, Linux commands, Shell Programming: Meaning and purpose of shell, Introduction to types of shell, The command line, standard input and standard output, redirection, pipes, filters special characters for searching files and pathnames., command separation & grouping, redirection, directory stack manipulation, processes, shell script-writing and executing, parameters & variables, keyword variables, Command line arguments, exit and exit status of a command. Logical operators for conditional execution. if, while, for and case control statements. The set and shift command.

4. Suggested Books and References

- [1] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 10th ed. Hoboken, NJ, USA: Wiley, 2018.
- [2] A. S. Tanenbaum and H. Bos, *Modern Operating Systems*, 5th ed. Boston, MA, USA: Pearson, 2023.
- [3] S. Das, *Your Unix/Linux: The Ultimate Guide*, 4th ed. New York, NY, USA: McGraw-Hill Education, 2018.
- [4] W. Shotts, *The Linux Command Line: A Complete Introduction*, 2nd ed. San Francisco, CA, USA: No Starch Press, 2019.
- [5] W. Stallings, *Operating Systems: Internals and Design Principles*, 9th ed. Boston, MA, USA: Pearson, 2018.
- [6] R. Love, *Linux Kernel Development*, 3rd ed. Boston, MA, USA: Addison-Wesley Professional, 2010.
- [7] C. Newham, *Learning the bash Shell: Automated Command Line Administration*, 4th ed. Sebastopol, CA, USA: O'Reilly Media, 2024

mcs-201 Web Development	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the fundamental syntax, data structures, execution environment, and modular file integration mechanics of server-side PHP scripting.
- Understand clean, functional, and object-oriented programming methodologies in PHP to write reusable and highly maintainable server-side components.
- Implement strict web data handling pipelines, ensuring secure input validation, structured state management (sessions/cookies), and advanced string parsing using regular expressions.
- Implement secure database communication protocols, emphasizing the absolute mitigation of SQL Injection vulnerabilities via PHP Data Objects (PDO) and prepared statements.
- Analyze architectural design paradigms including the Model-View-Controller (MVC) pattern, dependency orchestration with Composer, and stateless RESTful API development.

2. Course Outcomes (COs)

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

CO-1: Construct structural server-side web scripts utilizing standard PHP control flows, error handling frameworks, and advanced array manipulation matrices.

CO-2: Apply Object-Oriented Programming (OOP) concepts such as inheritance, visibility modifiers, and class structures to engineer scalable backend applications.

CO-3: Design secure web data state layers that effectively process form payloads, track persistence via cookies and sessions, and filter inputs to mitigate security vulnerabilities.

CO-4: Formulate secure database queries using PHP Data Objects (PDO), implementing transactional mechanics and parameter binding to build exploit-proof relational integrations.

CO-5: Assemble decoupled, modular web applications by integrating Composer package managers, implementing the MVC architecture.

3. Course Syllabus

mcs-201 Web Development	Duration: 3 hours Max Marks: 70
--	--

Unit 1:

Introduction to PHP runtime mechanics, web server interaction (Apache/Nginx), integration with HTML, syntax rules, and error handling configurations. Core Syntax & Control Structures, Working with data types, strict variable declarations, constants, complex expressions, operators, conditional execution (if-else, switch, match expressions), and iterative loops (for, while, foreach).

Modular Code Integration, Implementation of include, require, include_once, and require_once statements for clean file organization and separation of concerns. Defining and calling user-defined functions, handling positional and named parameters, defining strict return values, evaluating variable scopes, and utilizing anonymous/arrow functions. Understanding indexed vs. associative arrays, navigating multidimensional arrays, executing array traversal, array sorting, and applying built-in array manipulation functions.

Unit 2:

Object-Oriented PHP Foundations: Transitioning to OOP; declaring classes, initializing objects, defining visibility modifiers (public, private, protected), constructors, method handling, and basic inheritance. Secure Database Connectivity (PDO), Establishing secure connections to MySQL databases using PHP Data Objects (PDO); writing parameterized queries and executing Prepared Statements to completely eliminate SQL Injection (SQLi) risks. Data Mutations: Executing secure CRUD operations (Create, Read, Update, Delete), handling database transaction errors, and rendering query results cleanly into responsive web tables.

Unit 3:

Ecosystem Tooling: Introduction to Composer for dependency management, managing third-party libraries, and enforcing PSR (PHP Standards Recommendations) style guidelines and autoloading. Model-View-Controller (MVC) Pattern: Deconstructing the monolithic architecture; understanding the roles of Models (data logic), Views (user interface), and Controllers (request orchestration).

4. Suggested Books and References

[1] R. Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*, 6th ed. Sebastopol, CA, USA: O'Reilly Media, 2021.

[2] D. Sklar, *Learning PHP*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2016.

[3] C. McCool, *PHP Objects, Patterns, and Practice*, 6th ed. Berkeley, CA, USA: Apress, 2021.

[4] The PHP Documentation Group. *PHP Manual: Language Reference, PDO, and Session Handling*. [Online]. Available: php.net

[5] J. Lockhart, *Modern PHP: New Features and Good Practices*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2015.

[6] The OWASP Foundation. *OWASP Top 10 Web Application Security Risks*. [Online]. Available: owasp.org

mcs-202	Duration: 3 hours
Enterprise Networking and Cloud Networks	Max Marks: 70

1. Course Objectives

The primary objectives of this course are to:

- Understand the structural architecture of the OSI and TCP/IP reference models alongside the hierarchical design rules of enterprise campus networks.
- Understand the mathematical and operational mechanics of IP addressing, including high-density Variable Length Subnet Masking (VLSM) and modern IPv6 deployment schemas.
- Implement professional enterprise-tier routing and switching protocols, focusing on loop mitigation (STP), broadcast boundaries (VLANs).
- Examine modern Layer-7 applications, transport layer upgrades (QUIC/HTTP3), and cloud infrastructure networking concepts like Virtual Private Clouds (VPCs) and cloud edge security.
- Develop foundational troubleshooting capabilities using industry-standard simulation and emulation tools (Cisco Packet Tracer/GNS3) to build, secure, and defend hybrid infrastructure.

2. Course Outcomes (COs)

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

CO-1: Analyze hierarchical network frameworks and solve complex topological problems by engineering multi-tier VLSM IPv4 and stateless IPv6 addressing schemas.

CO-2: Configure, maintain, and troubleshoot enterprise switching fabrics by deploying Spanning Tree Protocol (STP), trunked Virtual Local Area Networks (VLANs), and VTP management domains.

CO-3: Construct scalable routing architectures by implementing static paths routing protocols to achieve optimal network convergence.

CO-4: Defend perimeter networks against unauthorized resource utilization by implementing Standard/Extended Access Control Lists (ACLs) and deploying structural NAT/PAT translations.

CO-5: Design and test hybrid corporate cloud environments by provisioning secure Virtual Private Clouds (VPCs), configuring cloud gateway route tables, and evaluating Software-Defined WAN (SD-WAN) edge rules.

3. Course Syllabus

mcs-202	Duration: 3 hours
Enterprise Networking and Cloud Networks	Max Marks: 70

Unit 1:

OSI Reference Model, TCP/IP stack configuration, and the Cisco 3-layer hierarchical campus design topology. Hubs, bridges, switches, and high-performance routers; transitions from legacy topologies to modern high-speed switched Ethernet environments.

IP Addressing Architecture, IPv4 addressing schemes; Classful vs. Classless Inter-Domain Routing (CIDR); Variable Length Subnet Masking (VLSM) and multi-tier subnetting. IPv6 representation, host address assignment, unicast, multicast, and unique local address constraints.

Unit 2:

Enterprise Switching Ecosystem, Spanning Tree Protocol (STP) and Rapid-STP (RSTP); full-duplex vs. half-duplex collision handling. Virtual Local Area Networks (VLANs); Relational Routing Architecture: Fundamental router operations, boot sequences, CLI executive privileges, configuration management, and baseline IOS security commands. Network Address Translation (NAT): Resolving public-private address depletion via Static NAT, Dynamic NAT, and Port Address Translation (PAT).

Unit 3:

Modern Layer-7 Infrastructure, Domain Name System (DNS) hierarchy; dynamic IP management via DHCP configuration; transport security models (TLS 1.3). Cloud Infrastructure Networks: Virtual Private Cloud (VPC) primitives; defining public and private cloud subnets, Internet Gateways (IGW), Virtual Private Gateways, route tables, and Cloud Access Control Lists (Security Groups). Multiprotocol Label Switching (MPLS) and Software-Defined WAN (SD-WAN) control plane architectures.

4. Suggested Books and References

[1] J. Odom, *CCNA 200-301 Official Cert Guide Library*, 1st ed. Indianapolis, IN, USA: Cisco Press, 2020.

[2] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*, 8th ed. Boston, MA, USA: Pearson, 2020.

[3] A. S. Tanenbaum, D. J. Wetherall, and N. Feamster, *Computer Networks*, 6th ed. Boston, MA, USA: Pearson, 2021.

[4] B. Piper and D. Clinton, *AWS Certified Advanced Networking Official Study Guide: Specialty Exam*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2018.

[5] Cisco Systems, *Cisco IOS IP Routing: OSPF Command Reference and Configuration Guide*. [Online]. Available: <https://cisco.com>

[6] Amazon Web Services, *Amazon VPC User Guide: Core Networking Infrastructure Components*. [Online]. Available: <https://amazon.com>

mcs-203 Programming in Java	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the core syntax of the Java programming language, including primitive data types, explicit type casting, operators, and control flow matrices.
- Understand the fundamental paradigms of Object-Oriented Programming (OOP) by creating structured classes, objects, parameterized constructors, and arrays.
- Implement safe data parsing routines using Java String manipulation techniques, `StringBuffer`, and memory-optimized `StringBuilder` objects.
- Apply code reusability strategies through method overloading, multi-level class inheritance hierarchies, and the execution of polymorphic super-class bindings.
- Develop scalable, fault-tolerant software frameworks by deploying decoupled interfaces, architectural packages, and robust exception-handling hierarchies.

2. Course Outcomes (COs)

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

CO-1: Construct syntax-compliant Java scripts that leverage strict variable typings, arithmetic operators, dynamic loops, and multi-dimensional array containers.

CO-2: Build production-grade class blueprints, utilizing parameterized constructors and explicit instance initializations via the `new` operator.

CO-3: Formulate memory-efficient text processing routines using specialized string constructor manipulations, index lookups, and dynamic buffer builders.

CO-4: Design deep inheritance structures and overloaded method layers, implementing the `super` token to orchestrate multi-level class constructor loops.

CO-5: Deploy modular software environments by enforcing isolated package namespace domains, extending functional multi-interfaces, and intercepting runtime anomalies with nested `try-catch` structures.

3. Course Syllabus

mcs-203 Programming in Java	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Java's Primitive Data Types, Literals, Variables, operators, Type conversion in Assignments, Cast, Operator Precedence, Expressions. Program Control Statements, if statement, Switch Statement, for Loop, While Loop, do-while Loop, Nested Loops. Introduction to Classes, Objects and Methods, Using Parameters, Constructors, Parameterized Constructors, The new operator. Arrays, Multidimensional Arrays.

Unit 2

String Fundamentals, The String Constructors, Three String-Related Language Features, The Length() Method, Obtaining the characters within a string, String comparison, using indexOf() and lastIndexOf(), Changing the case of characters within a string, String Buffer and String Builder.

Method Overloading, Overloading Constructors, Recursion, Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Super class constructors, Using super to Access Super class Members, Creating a Multilevel Hierarchy,

Unit 3

Interface Fundamentals, Creating an Interface, Implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Interfaces can be extended

Package Fundamentals, Packages and Member Access, Importing Packages, Static Import

Exception Handling, The Exception Hierarchy, Exception Handling Fundamentals, using multiple catch clauses, catching subclass Exceptions, try blocks can be nested, throwing an Exception

4. Suggested Books and References

- [1] H. Schildt, *Java: The Complete Reference*, 12th ed. New York, NY, USA: McGraw-Hill Education, 2021.
- [2] H. Schildt, *Java: A Beginner's Guide*, 9th ed. New York, NY, USA: McGraw-Hill Education, 2022.
- [3] J. Bloch, *Effective Java*, 3rd ed. Boston, MA, USA: Addison-Wesley Professional, 2018.
- [4] P. Deitel and H. Deitel, *Java How to Program, Early Objects*, 11th ed. Boston, MA, USA: Pearson, 2017.
- [5] Oracle Corporation. *The Java Tutorials: Core Java Classes and Language Basics*. [Online]. Available: <https://oracle.com>
- [6] Y. D. Liang, *Introduction to Java Programming and Data Structures, Comprehensive Version*, 12th ed. New York, NY, USA: Pearson, 2020.

mcs-204a 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

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

Unit 1

Software Process Models: Software Process, Generic Process Model – Framework Activity, Task Set and Process Patterns; Process Lifecycle, Prescriptive Process Models, Project Management, Component Based Development, Aspect-Oriented Software Development, Formal Methods, Agile Process Models – Extreme Programming (XP), Adaptive Software Development, Scrum, Dynamic System Development Model, Feature Driven Development, Crystal, Web Engineering.

Unit 2

Software Requirements: Functional and Non-Functional Requirements; Eliciting Requirements, Developing Use Cases, Requirement Analysis and Modelling; Requirements Review, Software Requirement and Specification (SRS) Document. 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, Risk Management, Risk Mitigation, Monitoring and Management (RMMM); Software Reliability.

Unit 3

Estimation and Scheduling of Software Projects: Software Sizing, LOC and FP based Estimations; Estimating Cost and Effort; Estimation Models, Constructive Cost Model (COCOMO), Project Scheduling and Staffing; Time-line Charts. Software Testing: Verification and Validation; Error, Fault, Bug and Failure; Unit and Integration Testing; White-box and Black-box Testing; Basis Path Testing, Control Structure Testing, Deriving Test Cases, Alpha and Beta Testing; Regression Testing, Performance Testing, Stress Testing.

Software Configuration Management: Change Control and Version Control; Software Reuse, Software Re-engineering, Reverse Engineering

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 (SWEBOK 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.

mcs-204b Data Structures and Algorithms	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- **Theoretical Foundations:** Master fundamental concepts of time and space complexity analysis.
- **Core Mechanisms:** Understand design and internal mechanics of linear data structures.
- **Algorithmic Selection:** Learn sorting, searching, and hashing paradigms to optimize retrieval.
- **Hierarchical Logic:** Explore structural properties of non-linear entities like trees and graphs.
- **Practical Solutions:** Solve complex computational problems through optimal algorithmic choices.

2. Course Outcomes (COs)

Upon completing this course, students will be able to:

CO1: Analyze execution time and space limits for recursive and non-recursive code.

CO2: Build linear operations using arrays, stacks, queues, and linked lists.

CO3: Compare computational overhead among classical sorting and searching algorithms.

CO4: Manipulate hierarchical datasets using binary search trees and AVL balanced rotations.

CO5: Traverse interconnected networks and compute minimal path frameworks via graph models.

CO6: Resolve data collisions using open and closed hashing collision-resolution policies.

3. Course Syllabus

mcs-204b Data Structures and Algorithms	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Introduction to Algorithms and Analysis: Fundamentals of algorithm analysis, Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth, Algorithm efficiency – best case, worst case, average case, Analysis of non-recursive and recursive algorithms.

Linear Data Structures: Array- 1D and 2D array, Stack - Applications of stack: Expression Evaluation - Conversion of Infix to postfix and prefix expression, Tower of Hanoi. Queue - Types of Queues: Circular Queue, Double Ended Queue (deQueue), Applications – Priority Queue using Arrays - List - Singly linked lists – Doubly linked lists - Circular linked lists,

Unit 2

Sorting and Search Techniques: Sorting Algorithms: Basic concepts, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Shell Sort, Heap Sort, Merge Sort, Counting Sort, External Sorting, Internal Sorting, Stable & Unstable Sorting. Searching: Linear Search, Binary Search.

Unit 3

Trees: Terminology, Binary Tree – Terminology and Properties, Tree Traversals, Expression Trees – Binary Search Trees – operations in BST – insertion, deletion, Searching. AVL Trees-Insertion, deletion and Rotation in AVL Trees,

Graphs & Hashing: Basic definition and Terminology – Representation of Graph – Graph Traversal: Breadth First Search (BFS), Depth First Search (DFS) - Minimum Spanning Tree: Prim's, Kruskal's- Single Source Shortest Path: Dijkstra's Algorithm. Hashing: Introduction, open hashing-separate chaining, closed hashing - linear probing, quadratic probing, double hashing, random probing, rehashing, Recent Trends in Data Structures and Algorithms

3. Suggested Books and References

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.
- [2] E. Horowitz, S. Sahni, and S. Anderson-Freed, *Fundamentals of Data Structures in C*, 2nd ed. Silicon Valley, CA, USA: Silicon Press, 2008.
- [3] M. A. Weiss, *Data Structures and Algorithm Analysis in C++*, 4th ed. Boston, MA, USA: Pearson, 2014.
- [4] R. Sedgewick and K. Wayne, *Algorithms*, 4th ed. Upper Saddle River, NJ, USA: Addison-Wesley, 2011.
- [5] G. F. Michael, R. Tamassia, and D. M. Mount, *Data Structures and Algorithms in C++*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2011.
- [6] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Boston, MA, USA: Addison-Wesley, 1983.

mcs-301	Duration: 3 hours
Mobile Application Development	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

mcs-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

mcs-302 Data Warehousing and Data Mining	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

Introduce the architectural principles of Data Warehousing (DWH) and Online Analytical Processing (OLAP).

Teach structural design techniques for data repositories using multi-dimensional modeling.

Build foundational knowledge of the ETL (Extract, Transform, Load) pipeline architecture.

Develop Understanding over data mining workflows and algorithmic pattern recognition.

Train students to implement classification, clustering, and association rule algorithms on enterprise datasets.

2. Course Outcomes (COs)

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

CO1: Differentiate between transactional database architectures (OLTP) and analytical data warehouses (OLAP).

CO2: Design multi-dimensional schemas (Star, Snowflake) to structurally represent complex enterprise business metrics.

CO3: Construct robust ETL frameworks to clean, transform, and aggregate raw operational records into optimized data warehouses.

CO4: Apply association rule algorithms and mining frameworks to discover hidden patterns in transactional data.

CO5: Evaluate and execute predictive classification and clustering models to group data and predict behavioral trends.

3. Course Syllabus

mcs-302	Duration: 3 hours
Data Warehousing and Data Mining	Max Marks: 70

Unit 1

Introduction to Data Warehousing, Definition, characteristics (subject-oriented, integrated, time-variant, non-volatile), and operational databases (OLTP) vs. Data Warehouses (OLAP). Warehouse Architecture, Operational data stores, data marts, metadata repositories, three-tier data warehouse architecture, and enterprise data models. Multi-Dimensional Modeling, Data cubes, dimensions, measures, Star Schema, Snowflake Schema, and Fact Constellation (Galaxy) Schema. OLAP Operations Drill-down, roll-up, slice, dice, and pivot operations.

Unit 2

The ETL Process, Data Extraction techniques, Transformation mapping (cleansing, sorting, deriving metrics), Loading strategies (initial load vs. incremental refresh), and Data Quality Management. Introduction to Data Mining, Definition, knowledge discovery process (KDD lifecycle), data mining functionalities, and major challenges in data mining. Using Weka for Data Mining. Data Preprocessing for Mining, Data cleaning, data integration, data reduction (sampling, dimensionality reduction), and data transformation (discretization and normalization).

Unit 3

Association Rule Mining, Market Basket Analysis, frequent itemsets, support, confidence, and lift metrics. Usage of Apriori and FP-Growth algorithm variants in Weka for business insights. Advanced Classification, Overview of decision tree induction (ID3/C4.5), Naïve Bayes classification, and performance metrics (Confusion Matrix, ROC curves). Using Arff files and Csv files in Weka for classification in business datasets. Advanced Clustering: Partitioning methods (K-Means, K-Medoids), hierarchical methods (AGNES, DIANA), and density-based methods (DBSCAN). Brief introduction to other clustering options in Weka. Outlier Analysis: Types of outliers, statistical outlier detection methods, and proximity-based outlier mining.

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, 2011.
- [2] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Indianapolis, IN, USA: John Wiley & Sons, 2013.
- [3] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. Boston, MA, USA: Pearson, 2018.
- [4] V. Rainardi, *Building a Data Warehouse: With Examples in SQL Server*, 1st ed. Berkeley, CA, USA: Apress, 2008

mcs-303a Business Data Analytics	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the corporate data landscape, aligning the business analytics taxonomy (descriptive, predictive, prescriptive) with organizational strategy and strict data governance principles.
- Understand data engineering pre-processing steps, including ingestion pipelines, structured data wrangling, outlier identification, and exploratory data profiling techniques.
- Apply visual design methodologies and interface metrics to synthesize interactive business dashboards and analytical chart portfolios.
- Implement advanced statistical modeling architectures—including linear/logistic regressions and longitudinal ARIMA models—to forecast macro trends and calculate discrete risk default boundaries.
- Evaluate supervised and unsupervised machine learning algorithms to build robust feature engineering workflows and segment enterprise behavioral customer fields.

2. Course Outcomes (COs)

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

CO-1: Construct unified data preparation workflows to ingest unstructured fields, isolate multi-variate statistical correlation patterns, and handle data anomalies safely.

CO-2: Design and deploy structural executive KPI dashboards and multivariate data plots adhering to strict professional visual design constraints.

CO-3: Formulate linear and logistic predictive regressions to evaluate corporate churn metrics, determine log-odds conversions, and optimize confusion matrix diagnostics.

CO-4: Model multi-component time-series forecasting pipelines using exponential smoothing and integrated moving average architectures to evaluate historical longitudinal data.

CO-5: Assemble prescriptive decision models under uncertainty (using EMV, payoff matrices, and tree configurations) and deploy optimized tree-based and K-means machine learning classifiers.

3. Course Syllabus

mcs-303a	Duration: 3 hours
Business Analytics	Max Marks: 70

Unit 1

Introduction to the business analytics taxonomy (descriptive, predictive, prescriptive); alignment of data analytics with corporate strategy; data quality, parsing, and governance principles. Data Preparation & Wrangling, Ingesting and transforming unstructured and structured business datasets; text parsing, handling missing data rows, outlier detection, Exploratory Data Analysis (EDA): Finding hidden distribution patterns, processing multi-variate statistical variables, and running variable correlation tests.

Unit 2

Data Visualization Paradigms: Visual design rules; constructing interactive visual charts, histograms, box-plots, and scatter plots; building structural executive dashboards to track corporate KPIs. Predictive Analytics, Forecasting, and Statistical Modeling, Linear Regression Analysis, Predicting corporate churn, default risks, and customer conversions using Logistic Regression; understanding log-odds, maximum likelihood estimation, and confusion matrix diagnostics. Time-Series Forecasting: Evaluating longitudinal historical metrics; component tracking (trend, seasonality, cyclical, random variations)

Unit 3

Prescriptive Analytics, decision-making under uncertainty, payoff matrices, expected monetary value (EMV), utility theory, and complex decision-tree architectures.

Supervised Machine Learning Workflows: Transitioning from traditional statistics to algorithmic ML; feature engineering pipelines, tree-based models (Random Forests, Gradient Boosting), and hyper-parameter validation loops.

Unsupervised Clustering for Market Segmentation: Grouping structural behavioral datasets; executing K-Means clustering, hierarchical clustering algorithms, and evaluating cluster distance matrix metrics for targeted geo-marketing campaigns.

4. Suggested Books and References

- [1] J. R. Evans, *Business Analytics*, 3rd ed. Boston, MA, USA: Pearson, 2020.
- [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] C. N. Knaflitz, *Storytelling with Data: A Data Visualization Guide for Business Professionals*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2015
- [4] C. O. Wilke, *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
- [5] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 6th ed. Hoboken, NJ, USA: John Wiley & Sons, 2021.
- [6] F. S. Hillier and M. S. Hillier, *Introduction to Management Science: A Modeling and Case Studies Approach with Spreadsheets*, 6th ed. New York, NY, USA: McGraw-Hill Education, 2019.

mcs-303b Artificial Intelligence and Heuristics	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the theoretical foundations of symbolic artificial intelligence, including agent architectures, environmental properties, and computational rationality.
- Understand the mathematical and space/time complexity bounds of both uninformed (BFS, DFS, UCS) and informed heuristic search strategies.
- Apply adversarial game-tree optimization algorithms and constraint satisfaction workflows to resolve competitive or resource-bound problems.
- Implement formal propositional logic systems, syntactic semantics, and resolution proofs using Conjunctive Normal Form (CNF) abstractions.
- Evaluate automated planning domain structures (PDDL) and rule-based expert systems to construct autonomous decision-making inference engines.

2. Course Outcomes (COs)

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

CO-1: Construct formal state-space formulations and analyze the algorithmic constraints of tree and graph searches within dynamic problem domains.

CO-2: Design and optimize localized heuristic search routines, utilizing meta-heuristics like Simulated Annealing and Genetic Algorithms to bypass local maxima plateaus.

CO-3: Formulate competitive decision engines using Minimax architectures, Expectiminimax stochastic variables, and Alpha-Beta pruning to minimize search tree depth.

CO-4: Resolve multi-variable constraint problems by applying backtracking paradigms and evaluating domain constraints under bounded conditions.

CO-5: Transcribe abstract domain definitions into formal propositional expressions to execute Conjunctive Normal Form (CNF) proofs, PDDL planning graphs, and forward/backward rule-based expert inferences.

3. Course Syllabus

mcs-303b Artificial Intelligence and Heuristics	Duration: 3 hours Max Marks: 70
--	--

Unit 1

Foundations of Classical AI: The Turing test, the physical symbol system hypothesis, rationality, agent architectures, and environmental properties. State-Space Search, Graph and tree search formulations; analyzing space/time complexities of Breadth-First Search (BFS), Depth-First Search (DFS), Depth-Limited Search, and Uniform Cost Search (UCS). Informed (Heuristic) Search: Evaluation functions and heuristic design; Solving high-density constraint spaces using Hill-Climbing (and mitigating local maxima/plateaus), Simulated Annealing, Local Beam Search, and Genetic Algorithms.

Unit 2

Game Theory & Competitive Environments, Formulating zero-sum, perfect-information games; the Minimax algorithm for optimal decision-making. Optimizing minimax search trees via Alpha-Beta Pruning; establishing evaluation functions and cutoff tests for deep game trees. Stochastic & Imperfect Games: Handling uncertainty using Expectiminimax for games of chance; introduction to game theory concepts like Nash Equilibrium. Constraint Satisfaction Problems (CSPs): Defining states as collections of variables with domain constraints; solving CSPs using Backtracking Search.

Unit 3

Knowledge Representation, Logic Systems, & Automated Reasoning, Propositional Logic Syntax, semantics, truth tables, entailment, validity, satisfiability, and inference rules (Modus Ponens); converting expressions to Conjunctive Normal Form (CNF) for resolution proofs. Classical Planning: Planning Domain Definition Language (PDDL); state-space planning, planning graphs, and hierarchical task networks. Expert Systems: Rule-based knowledge architectures, forward/backward execution engines, handling uncertainty via certainty factors, and building corporate inference engines.

4. Suggested Books and References

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2020.
- [2] N. J. Nilsson, *Artificial Intelligence: A New Synthesis*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 1998.
- [3] E. Rich and K. Knight, *Artificial Intelligence*, 3rd ed. New York, NY, USA: McGraw-Hill Education, 2008.
- [4] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 2004.
- [5] D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, 3rd ed. Cambridge, UK: Cambridge University Press, 2023.
- [6] Association for the Advancement of Artificial Intelligence (AAAI). *AAAI Conference on Artificial Intelligence Resources*. [Online]. Available: aaai.org

mcs-304a Data Analytics and Insights	Duration: 3 hours Max Marks: 70
---	--

1. Course Objectives

- Formalize the standardized lifecycle models of data analytics used to solve structured business problems.
- Teach the classification and application of different analytical types (Descriptive, Diagnostic, Predictive, Prescriptive).
- Build foundational skills in designing measurable business KPIs, metrics, and data tracking frameworks.
- Develop professional competency in data auditing, anomaly discovery, and structural validation workflows.
- Train students to craft compelling, evidence-based business recommendations and data-driven narratives.

2. Course Outcomes (COs)

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

CO1: Map a complex, ambiguous corporate problem to a structured, repeatable analytical framework like CRISP-DM.

CO2: Formulate and construct measurable business KPIs and cohort segments.

CO3: Perform formal data profiling and quality audits to evaluate structural data pipelines.

CO4: Apply root-cause analysis methodologies to isolate and diagnose the drivers behind sudden shifts in business metrics.

CO5: Author professional data-driven narratives and executive slide-decks that clearly articulate technical findings to non-technical business leaders.

3. Course Syllabus

mcs-304a Data Analytics and Insights	Duration: 3 hours Max Marks: 70
---	--

Unit 1

Distinguishing Data Analytics from Data Science and Data Engineering. The value-complexity ladder of analytics. Introduction to Jira / Trello Agile Project Management and Kanban boards. The CRISP-DM (Cross-Industry Standard Process for Data Mining) lifecycle and OSEMN (Obtain, Scrub, Explore, Model, iNterpret) frameworks and case studies.

Problem Scoping & Translation: Mapping vague business problems to precise analytical queries. Defining target variables, objectives, and project constraints.

Unit 2

Descriptive & Diagnostic Deep-Dives: Conducting operational drill-downs, establishing cohort analysis frameworks, and using segmentation to isolate driver variables. Using Tableau for Descriptive & Diagnostic Deep-Dives, build interactive dashboards with drill-down capabilities.

Anomalies and Outlier Auditing: Distinguishing true data anomalies (errors) from extreme business events (e.g., holiday sales spikes). Impact of outliers on baseline business metrics.

Unit 3

Root-Cause Analysis (RCA): Applying structural frameworks (The 5 Whys, Fishbone Diagrams) to explain metric shifts observed in data reporting. Constructing executive summaries, action-oriented bullet points, and business case justifications. Create action-oriented executive summaries and business infographics using Canva / Infogram

Visual Presentation Frameworks: Storyboarding analytical findings, structuring reports for non-technical stakeholders, and matching report detail to executive tiers.

4. Suggested Books and References

- [1] N. P. Padhy, *Data Analytics Basics: A Beginner's Guide to Data Analysis*. Boca Raton, FL, USA: CRC Press.
- [2] B. Marr, *Key Performance Indicators (KPI): The 75+ Measures Every Manager Needs to Know*. London, UK: Financial Times Prentice Hall.
- [3] A. Croll and B. Yoskovitz, *Lean Analytics: Use Data to Build a Better Startup Faster*. Sebastopol, CA, USA: O'Reilly Media.
- [4] B. Marr, *Data Strategy: How to Profit from a World of Big Data, Analytics and the Internet of Things*. London, UK: Kogan Page.
- [5] C. N. Knafllic, *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Hoboken, NJ, USA: Wiley.

mcs-304b 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

mcs-304b 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

mcs-401	Duration: 3 hours
JavaScript with UI /UX User Centric Design Principles	Max Marks: 70

1. Course Objectives

The primary objectives of this course are to:

- Understand the collaborative intersection between visual UI/UX product pipelines, client-side JavaScript architecture, and user-centric design theories.
- Understand the application of core user-centric methodologies, including user research, persona modeling, heuristic evaluation, and responsive design systems.
- Apply semantic HTML5 and vanilla JavaScript syntax rules to engineer clean, responsive, and dynamic Document Object Model (DOM) logic.
- Implement non-blocking, asynchronous web services using the Fetch API to populate interactive user interfaces with real-time cloud data arrays.
- Evaluate digital interface accessibility metrics against strict WCAG standards to deploy inclusive, highly performant component systems.

2. Course Outcomes (COs)

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

CO-1: Construct high-fidelity, responsive user interface wireframes and interactive prototypes in modern design tools following user-centric usability heuristics.

CO-2: Design and troubleshoot client-side scripts using JavaScript control flows, complex array data structures, and functional execution scopes.

CO-3: Formulate real-time user interface mutations by programmatically binding listeners to handle keyboard, form validation, and pointer event states.

CO-4: Model asynchronous API data extraction layers to fetch external JSON payloads and parse them into localized interactive elements.

CO-5: Assemble a modular, fully accessible frontend component library (modals, sliders, tabs) that strictly satisfies WCAG color contrast and screen-reader requirements.

3. Course Syllabus

mcs-401	Duration: 3 hours
JavaScript with UI /UX User Centric Design Principles	Max Marks: 70

Unit 1

UI/UX Design Fundamentals, Wireframing, and Accessibility, User Experience (UX) Core Foundations Understanding user-centric design methodologies, user research, persona creation, and mapping user journey blueprints. User Interface (UI) Visual Principles, Layout grids, alignment, negative space, visual hierarchy, typography, and cohesive digital design color theory. Wireframing & Interactive Prototyping using design tools (Figma / Adobe XD); designing responsive grids for mobile, tablet, and desktop viewports.

Unit 2

Introduction to Core JavaScript Logic and Client-Side Interactivity, JavaScript Language Ingestion (ES6+), Scripting environment setups, variables, primitive types, array structures, literal strings, loops, and conditional control blocks. Functional Programming & Variable Scopes: Defining and executing user-defined functions, handling parameters, and understanding local vs. global lexical execution scopes. The Document Object Model (DOM) Architecture, DOM tree; updating inner content, altering CSS layout styles dynamically, and appending/removing nodes. Interactive Event Handling Systems, capturing user interface events (click, submit, input, keydown, scroll), registering event listeners, and preventing standard browser event defaults.

Unit 3

Component-Driven Frontend Engineering, UI components; building modular tabs, responsive navigation drawers, modal window overlays, and sliding asset carousels. Form Validation & Error Notification Interfaces: Processing user inputs; engineering client-side data validation algorithms, displaying inline error handling warnings, and creating accessible data response fields. Micro-interactions & Functional Transitions: Animating state mutations; applying fluid CSS transitions alongside JavaScript event triggers to build micro-interactions that elevate UX satisfaction metrics.

4. Suggested Books and References

Standard Textbooks

- [1] J. Robbins, *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
- [2] M. Haverbeke, *Eloquent JavaScript: A Modern Introduction to Programming*, 4th ed. San Francisco, CA, USA: No Starch Press, 2024.
- [3] W. Albert and T. Tullis, *Measuring the User Experience: Collecting, Analyzing, and Presenting UX Metrics*, 3rd ed. Cambridge, MA, USA: Morgan Kaufmann, 2022.
- [4] D. Flanagan, *JavaScript: The Definitive Guide*, 7th ed. Sebastopol, CA, USA: O'Reilly Media, 2020.
- [5] D. Norman, *The Design of Everyday Things: Revised and Expanded Edition*, 2nd ed. New York, NY, USA: Basic Books, 2013.

mcs-403a Predictive Business 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

mcs-403a Predictive Business Analytics	Duration: 3 hours Max Marks: 70
---	--

Unit 1

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: Demystifying black-box machine learning models for corporate stakeholders; computing global and local feature contributions using SHAP (Shapley Additive exPlanations) and LIME metrics.

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.
- [6] Journal of Business Analytics. *Taylor & Francis Group Research Archive*. [Online]. Available: tandfonline.com

mcs-403b Generative AI & Prompt Engineering	Duration: 3 hours Max Marks: 70
--	--

1. Course Objectives

The primary objectives of this course are to:

- Understand the paradigm shift from classical symbolic AI to deep autoregressive architectures, evaluating transformer attention blocks and model lifecycles.
- Identify and mitigate systemic LLM vulnerabilities, including contextual drift, alignment biases, and data hallucinations.
- Apply structured prompt framework methodologies, such as the CO-STAR architecture, to enforce strict output schemas and semantic styles.
- Implement stateful conversational agent core structures by designing system message protocols, interaction memory trees, and multi-turn workflows.
- Evaluate advanced reasoning heuristics, including Few-Shot learning, Chain-of-Thought, and recursive error-correction tracking loops to optimize model throughput.

2. Course Outcomes (COs)

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

CO-1: Analyze transformer tokenization pipelines and execution mechanics to diagnose and neutralize hallucinations or toxic outputs in generative systems.

CO-2: Construct context-conditioned prompt skeletons using the components of instructions, contextual boundaries, and structural formatting directives.

CO-3: Design and deploy deterministic instruction templates leveraging the CO-STAR architecture to generate predictable, type-safe data shapes.

CO-4: Formulate stateful chatbot memory arrays and conversational routing architectures capable of maintaining long-context multi-turn interactive flows.

CO-5: Assemble advanced programmatic reasoning tracks—including Few-Shot demonstrations, Chain-of-Thought branches, and automated refinement loops—to simulate specialized technical personas.

3. Course Syllabus

mcs-403b	Duration: 3 hours
Generative AI & Prompt Engineering	Max Marks: 70

Unit 1

Introduction to generative modelling paradigms; historical evolution of symbolic AI to deep neural networks; AI literacy within global enterprise digital transformations. High-level analysis of transformer architectures, tokenization parameters, contextual attention spans, and pre-training/fine-tuning lifecycles; diagnosing systemic vulnerabilities: data hallucinations, contextual drift, algorithmic biases, and toxic outputs.

Foundations of Prompt engineering; human-in-the-loop interaction parameters; identifying prompt components: instructions, contextual boundaries, input data frames, and formatting directives.

Unit 2

Structured Prompt Frameworks, The CO-STAR Architecture Blueprint, Deconstructing the CO-STAR prompting matrix; establishing deterministic Context, defining clear operational Objectives, enforcing rigorous semantic Styles/Tones, mapping task-specific execution tracks (Truth/Tasks), profiling target Audiences, and formatting predictable structural Responses. Functional prompt layout templates; implementing role assignment blueprints, persona simulations, instruction-based boundaries, and adaptive context injection patterns.

Unit 3

Modelling high-capacity specialized personas; configuring LLMs to simulate technical roles: Automated Educators, Academic Researchers, Enterprise Business Analysts, Software Engineers, and Strategic Consultants. Heuristic Reasoning Architectures, Bypassing base model logical limitations; implementing zero-shot vs. Few-Shot Learning demonstrations; engineering sequential Chain-of-Thought (CoT) reasoning networks and Tree-of-Thought decision tree algorithms. Iterative Optimization & Debugging: Automated prompt refinement loops; debugging runtime parsing exceptions, compiling recursive error-correction feedback tracking loops, and optimizing inference speeds.

4. Suggested Books and References

- [1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Hoboken, NJ, USA: Pearson, 2023.
- [2] T. Sanamrad and B. Arasteh, *Generative AI with Python: Foundations, Models, and Practical Applications*, 1st ed. New York, NY, USA: Packt Publishing, 2024.
- [3] J. Toro and J. Cuartas, *Advanced Prompt Engineering: Architecting Production-Grade Prompts for Language Models*, 1st ed. Birmingham, UK: Packt Publishing, 2024.
- [4] N. Gift and A. Deza, *Practical MLOps: How to Get Production Machine Learning Done*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2020.
- [5] OpenAI Applied Engineering Group. *Prompt Engineering Guide for Production Language Architectures*. [Online]. Available: [openai.com](https://openai.com/prompt-engineering)
- [6] Anthropic AI Research. *System Prompts, Context-Window Architectures, and Model Compliance Manuals*. [Online]. Available: anthropic.com

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

1. Course Objectives

The primary objectives of this course are to:

- Understand the architectural differences between transactional databases (OLTP) and analytical data frameworks (OLAP) within enterprise systems.
- Understand Kimball dimensional modeling methodologies to design robust, clean, and scalable corporate data star schemas.
- Apply multi-dimensional OLAP cube operations (Slice, Dice, Roll-up, Drill-down) to organize and pre-calculate massive business metrics portfolios.
- Implement Extract, Transform, and Load (ETL) pipelines to automate historical data migrations into structured staging layers.
- Evaluate data visualization design constraints and semantic access parameters to deliver highly secure, accessible, and high-performance corporate KPI dashboards.

2. Course Outcomes (COs)

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

CO-1: Construct unified corporate star schemas by declaring business grains, calculating fact models, and configuring slowly changing dimensions (SCDs).

CO-2: Design and deploy multi-dimensional OLAP cubes utilizing optimized storage models (MOLAP/ROLAP) to handle high-density enterprise queries.

CO-3: Formulate explicit Multi-Dimensional Expressions (MDX) scripts to slice data metrics along specific analytical axes and compile computed members.

CO-4: Model automated end-to-end ETL ingestion workflows that extract raw operational transactions and load validated records safely into target repositories.

CO-5: Assemble professional dashboard presentation systems integrated directly with multi-dimensional storage grids under tight role-based row security rules.

3. Course Syllabus

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

Unit 1

Evolution of BI architectures; operational data stores (ODS) vs. corporate data warehouses; Online Transaction Processing (OLTP) vs. Online Analytical Processing (OLAP). Designing enterprise data pipelines; the 4-step dimensional design process (Select the Business Process, Declare the Grain, Identify the Dimensions, Identify the Facts). Fact Table Architecture, Dimension Table Engineering; surrogate keys vs. natural keys; Star schemas vs. Snowflake schemas.

Unit 2:

OLAP Architecture Topologies, Comparative analysis of storage architectures: Multidimensional OLAP (MOLAP), Relational OLAP (ROLAP), and Hybrid OLAP (HOLAP). Defining unified dimensional spaces, measure groups, fact-dimension relationships, and attribute hierarchies. Navigating data cubes via logical analytical operations: **Slice** (isolating single dimensions), **Dice** (selecting sub-cubes), **Roll-up** (data aggregation up hierarchies), **Drill-down** (navigating to granular data blocks), and **Pivot** (re-orienting data views). Designing pre-calculated aggregation pathways to accelerate query speeds. Multi-Dimensional Expressions (MDX) Basics, execute analytical slice-and-dice operations using MDX commands.

Unit 3

Extract, Transform, and Load (ETL), Building data ingestion paths; extracting data from heterogeneous source systems, staging area data cleansing, parsing strings, handling missing values, and scheduling automated batch transformations. Business Intelligence Dashboard, Interface design metrics for executive management; identifying tracking indicators: Key Performance Indicators (KPIs), metrics, scorecards, and visual tracking trends. Visual Representation Design Rules, Cognitive visual design frameworks; constructing responsive layout systems.

4. Suggested Books and References

- [1] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Indianapolis, IN, USA: John Wiley & Sons, 2013.
- [2] E. Turban, R. Sharda, and D. Delen, *Business Intelligence, Analytics, and Data Science: A Managerial Perspective*, 4th ed. Boston, MA, USA: Pearson, 2018.
- [3] C. N. Knaflic, *Storytelling with Data: A Data Visualization Guide for Business Professionals*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.
- [4] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan Kaufmann, 2011.
- [5] Microsoft Engineering Group. *SQL Server Analysis Services (SSAS) Multi-dimensional Cube Modeling Design Manual*. [Online]. Available: microsoft.com
- [6] Tableau Software Team. *Enterprise Analytics Delivery, Visual Best Practices, and Dashboard Optimization Frameworks*. [Online]. Available: tableau.com

mcs-404b	Duration: 3 hours
Deep Learning & Neural Network Architectures	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

mcs-404b Deep Learning & Neural Network Architectures	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; Wasserstein loss tracking rules.

4. Suggested Books and References

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MCS 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>