GURU KASHI UNIVERSITY

Master of Computer Applications

(MCA)

Session: 2026–2027

Faculty of Computing

GRADUATE ATTRIBUTES OF THE PROGRAMME

Graduate Attributes represent the capabilities that the graduates of the MCA programme are expected to acquire. They define the learning outcomes for the master's degree.

Type of Learning Outcomes	**The Learning Outcomes Descriptors**
Disciplinary / Interdisciplinary Knowledge	Apply advanced knowledge in programming, software development, operating systems, computer graphics, AI, and cloud computing.
Design	Design and manage complex systems using data structures, algorithms, database technologies, and web/mobile development platforms.
Integration	Integrate interdisciplinary technologies such as IoT, data mining, machine learning, and big data analytics to solve real-world problems.
Research	Conduct research using scientific methodolo-gies and contribute to innovative computing solutions for business and societal needs.
Generic Learning Outcomes	Exhibit critical thinking, problem-solving, and decision-making skills in multidisciplinary environments.
Ethical Responsibility	Demonstrate ethical responsibility, teamwork, adaptability, and commitment to lifelong learning.
Leadership	Apply leadership and entrepreneurial skills to create IT-based solutions with societal relevance.

PROGRAMME STRUCTURE OF MASTER OF COMPUTER APPLICATION

Semester 1st

Code	Course Title	Type	L	T	P	Cr	Int	Ext	Total
MCA1400	Object Oriented Programming using C++	Core	3	0	0	3	30	70	100
MCA1401	Advanced Database Management Systems	Core	3	0	0	3	30	70	100
MCA1402	Object-Oriented Programming using C++ Lab	Core	0	0	2	1	30	70	100
MCA1403	Advanced Database Management System Lab	Core	0	0	2	1	30	70	100
MCA1404	Advanced Computer Architecture	Core	4	0	0	4	30	70	100
MCA1405	Software Project Management	Core	4	0	0	4	30	70	100
IKS0016	Indian Knowledge System	Indian Knowledge System	4	0	0	4	30	70	100
Discipline Elective I (Any one of the following)									
MCA1406	Cloud Computing	Discipline Elective I	4	0	0	4	30	70	100
MCA1407	Big Data	Discipline Elective I	4	0	0	4	30	70	100
MCA1408	Machine Learning	Discipline Elective I	4	0	0	4	30	70	100
		Total	**22**	**0**	**4**	**24**	**240**	**560**	**800**

Semester 2nd

Code	Course Title	Type	L	T	P	Cr	Int	Ext	Total
MCA2450	Advanced Data Structures & Algorithms	Core	3	0	0	3	30	70	100
MCA2451	Programming using Python	Core	3	0	0	3	30	70	100
MCA2452	Advanced Data Structure & Algorithms Lab	Core	0	0	2	1	30	70	100
MCA2453	Programming using Python Lab	Core	0	0	2	1	30	70	100
MCA2454	Advanced Operating System	Core	4	0	0	4	30	70	100
MCA2455	Advanced Computer Graphics	Core	3	0	0	3	30	70	100
MCA2456	Advanced Computer Graphics Lab	Core	0	0	2	1	30	70	100
MCA2457	Advance Web Technologies	Employability	0	0	4	2	30	70	100
Discipline Elective II (Any one of the following)									
MCA2458	IoT and Its Applications	Discipline Elective II	4	0	0	4	30	70	100
MCA2459	Digital Image Processing	Discipline Elective II	4	0	0	4	30	70	100
MCA2460	Data Warehousing and Data Mining Techniques	Discipline Elective II	4	0	0	4	30	70	100
		Total	**17**	**0**	**10**	**22**	270	630	900

PROGRAMME LEARNING OUTCOMES (MCA)

Master of Computer Application degree is awarded to students who have demonstrated the achievement of the outcomes located at level 4.5:

PSO	Element of the Descriptor	Programme Learning Outcomes relating to MCA
PSO1	Knowledge and understanding	Understand research methodology and software engineering practices to manage and execute complex projects.
PSO1	Knowledge and understanding	Demonstrate advanced knowledge in programming paradigms, data structures, algorithms, database management, computer architecture, operating systems, and emerging technologies such as AI, IoT, and mobile computing.
PSO1	Knowledge and understanding	Acquire deep insight into theoretical foundations of computer science such as Theory of Computation and Software Quality Assurance.
PSO1	Knowledge and understanding	Explore current trends in information systems and their applications in industry and academia.
PSO2	General, technical and professional skills required to perform and accomplish tasks	Design, develop, and deploy advanced software systems using modern tools, platforms, and frameworks.
PSO2	General, technical and professional skills required to perform and accomplish tasks	Conduct technical research and apply statistical and analytical methods to solve real-world IT problems.
PSO2	General, technical and professional skills required to perform and accomplish tasks	Perform system analysis, manage projects, and apply professional documentation and quality assurance standards.
PSO3	Application of knowledge and skills	Solve complex computing problems by integrating theoretical knowledge with practical implementation. Apply machine learning and data analytics to drive decision-making and business intelligence.
PSO4	Generic learning Outcomes	Exhibit leadership, collaboration, and problem-solving in multi-disciplinary teams. Engage in self-directed learning and stay updated with evolving technologies.
PSO5	Constitutional, humanistic, ethical, and moral values	Practice ethical computing, data privacy, and responsible use of information technology. Promote inclusivity, diversity, and sustainability in digital solutions.
PSO6	Employability and job-ready skills, and entrepreneurship skills and capabilities/qualities and mindset	Be equipped for roles such as software engineer, data analyst, AI/ML developer, cybersecurity specialist, network administrator, and cloud engineer. Possess strong entrepreneurial and innovation-driven mindset supported by project and internship experience.
PSO7	Credit requirements	90 Credits
PSO8	Entry requirements	Bachelor's Degree with Honours/ Honours with Research in Information Technology or Computer Applications as a major course or BCA with Honors/Honours with Research (4 years) OR 1 year PG Diploma in Computer Applications/Information Technology with at least 45% marks or Equivalent CGPA in aggregate, after 3 years Bachelor's Degree.

Semester 3rd

Code	Course Title	Type	L	T	P	Cr	Int	Ext	Total
MCA3500	Artificial Intelligence	Core	4	0	0	4	30	70	100
MCA3501	Research Methodology	Core	4	0	0	4	30	70	100
MCA3502	Network Administration	Employability	0	0	4	2	30	70	100
MCA3503	Dissertation (PhaseI)	Research Skills	0	0	0	12	30	70	100
		Total	**8**	**0**	**4**	**22**	**120**	**280**	**400**

Semester 4th

Code	Course Title	Type	L	T	P	Cr	Int	Ext	Total
MCA4550	Java Programming	Core	3	0	0	3	30	70	100
MCA4551	Java Programming Lab	Core	0	0	2	1	30	70	100
MCA4552	Mobile Application Development	Employability	0	0	4	2	30	70	100
MCA4553	Dissertation (Phase II)	Research Skills	0	0	0	12	30	70	100
Discipline Elective III (Any one of the following)									
MCA4554	Theory of Computation	Discipline Elective III	4	0	0	4	30	70	100
MCA4555	Principle & Practices of Management	Discipline Elective III	4	0	0	4	30	70	100
MCA4556	Software Testing and Quality Assurance	Discipline Elective III	4	0	0	4	30	70	100
		Total	**7**	**0**	**6**	**22**	**150**	**350**	**500**
		Grand Total	**54**	**0**	**24**	**90**	**780**	**1820**	**2600**

SEMESTER-I

Course Title: Object Oriented Programming using C++	**L**	**T**	**P**	**Cr.**
Course Code: MCA1400	3	0	0	3

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Explain the fundamentals of C++ programming, program structure, and core object-oriented programming concepts such as encapsulation, abstraction, inheritance, and polymorphism.

CO2. Apply classes, objects, functions, constructors, destructors, operator overloading, pointers, references, and dynamic memory management to solve programming problems.

CO3. Analyze inheritance, polymorphism, static binding, dynamic binding, virtual functions, abstract classes, and interfaces in object-oriented program design.

CO4. Develop C++ programs using streams, file handling, exception handling, templates, Standard Template Library components, and basic modern C++ features.

UNIT I **15 Hours**

Programming Basics: Introduction to programming, programming paradigms, programming languages and their types. Brief review of C programming structure, execution flow of a C program, directives, and basic input/output.

Introduction to C++: Structure of a C++ program, header files and source files, namespace concept, input/output using `cin` and `cout`, data types, control structures, operators, expressions, and comparison of C and C++.

Object-Oriented Programming Concepts: Meaning and need of object-oriented programming, advantages and applications of OOP, core OOP principles: encapsulation, abstraction, inheritance, and polymorphism.

Classes and Objects: Class definition, objects, access modifiers, data members, member functions, inline functions, scope resolution operator, passing parameters to a function by value, by address, and by reference.

Constructors, Destructors, Arrays, and Strings: Default constructor, parameterized constructor, copy constructor, constructor overloading, destructors, arrays as class members, arrays of objects, C-style strings, and C++ string class.

UNIT II **10 Hours**

Functions and Object Handling: Function overloading, objects as parameters, returning objects from functions, static data members and static member functions, constant data members and constant member functions.

Operator Overloading: Operator functions, member and non-member operator functions, friend functions, overloading unary and binary operators, arithmetic operators, relational operators, assignment operators, equality operators, subscript operator overloading, type-conversion operators: primitive to object, object to primitive, and object to object. Advantages, limitations, and misuse of operator overloading.

Pointers and References: Pointer concept, address-of operator, pointer initialization, pointer arithmetic, pointer to an array, array of pointers, pointer to constant, constant pointer, void pointer, null pointer, dangling pointer, references versus pointers, and `const` correctness.

Dynamic Memory Management: Dynamic memory allocation using `new` and `delete`, `this` pointer, object copying, copy constructor, assignment operator, introduction to resource safety, RAII concept, and basic awareness of smart pointers such as `unique_ptr` and `shared_ptr`.

UNIT III **10 Hours**

Inheritance: Concept of inheritance, need of inheritance, protected access modifier, modes of inheritance: public, private, and protected. Types of inheritance: single, multilevel, hierarchical, multiple, and hybrid inheritance.

Constructors and Inheritance: Constructor and destructor execution in inheritance, constructor calls in base and derived classes, function overriding, member hiding, multiple inheritance, multipath inheritance, ambiguity problems, and their solutions.

Polymorphism: Compile-time and run-time polymorphism, static binding and dynamic binding, virtual functions, pure virtual functions, virtual destructors, abstract classes, and interfaces.

UNIT IV **10 Hours**

Streams and File Handling: Stream classes, standard input/output streams, text and binary files, file input/output using stream classes, file pointers, error streams, random file access, manipulators, and overloading insertion and extraction operators.

Exception Handling: Error handling, exception handling mechanism, `try`, `throw`, and `catch`, throwing and catching exceptions, multiple catch blocks, custom exceptions, and built-in exceptions.

Templates and Standard Template Library: Function templates, class templates, template declaration and usage, introduction to the Standard Template Library, containers such as `vector`, `list`, `map`, and `set`, iterators, algorithms, string objects, hashes, and iostreams.

Modern C++ Basics: Basic use of `auto`, `nullptr`, range-based `for` loop, and conceptual introduction to Rule of Three and Rule of Five for safe object copy-ing and resource management.

Transactional Mode: Lecture method, e-team teaching, video-based learning, demonstration, peer discussion, open talk, cooperative teaching, flipped teaching, collaborative learning, and hands-on programming practice.

Suggested Readings:

- Kamthane, A. (2012). *Programming in C++* (2nd ed.). Pearson Education India. https://books.google.com/books?isbn=9788131760696
- Salaria, R. S. (2016). *Mastering Object-Oriented Programming with C++*. Khanna Publishing House. https://books.google.com/books?isbn=9789381068205
- Balagurusamy, E. *Object-Oriented Programming with C++*. McGraw-Hill Education. https://books.google.com/books?isbn=9781259029936
- Lippman, S. B., Lajoie, J., & Moo, B. E. *C++ Primer*. Addison-Wesley. https://books.google.com/books?isbn=9780321714114
- Stroustrup, B. *Programming: Principles and Practice Using C++*. Addison-Wesley. https://books.google.com/books?isbn=9780321992789

Course Title: Advanced Database Management System	**L**	**T**	**P**	**Cr.**
Course Code: MCA1401	3	0	0	3

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Explain advanced database architectures, data models, relational systems, NoSQL databases, and cloud database concepts.

CO2. Design normalized relational database schemas using ER/EER modeling, functional dependencies, and higher normal forms.

CO3. Write and manage advanced SQL queries, stored procedures, triggers, cursors, transactions, and concurrency-control mechanisms.

CO4. Analyze indexing, query optimization, database tuning, distributed databases, data warehousing, database security, backup, and recovery techniques.

UNIT I: Database Architectures, Data Models, and Modern Database Systems 15 Hours

Database System Architectures: Centralized database architecture, client-server database architecture, distributed database architecture, parallel database architecture, three-tier database architecture, and basic database deployment models.

Advanced Data Models: Relational model review, hierarchical model, network model, object-oriented data model, object-relational data model, semi-structured data model, and document-oriented data model.

Big Data and NoSQL Databases: Introduction to big data, limitations of traditional relational databases for large-scale and unstructured data, NoSQL database characteristics, types of NoSQL databases: key-value, document, column-family, and graph databases. Use cases of relational, NoSQL, and hybrid database systems.

Modern Database Trends: Introduction to cloud databases, database-as-a-service, distributed storage, scalable database systems, and basic awareness of NewSQL and vector databases.

UNIT II: Advanced SQL, PL/SQL, Transactions, and Concurrency Control 10 Hours

Advanced SQL Queries: Subqueries, correlated subqueries, joins, outer joins, set operations, aggregate functions, grouping, nested queries, views, and in-dexing basics for query support.

Stored Procedures and Database Programming: Stored procedures, functions, parameters, execution, management, advantages, and limitations. Introduction to PL/SQL or equivalent procedural SQL.

Triggers and Cursors: Definition and types of triggers, trigger execution, use cases of triggers, cursors, cursor operations, and applications of cursors in database processing.

Transaction Management and Concurrency Control: ACID properties, transaction states, commit and rollback, isolation levels, serializability, locking, deadlock handling, timestamp-based protocols, and concurrency-control techniques.

UNIT III: Database Design, Normalization, Indexing, and Query Optimization 10 Hours

Advanced ER and EER Modeling: Entity-relationship modeling, extended ER modeling, subclasses, superclasses, specialization, generalization, aggregation, inheritance, and mapping EER diagrams to relational schemas.

Normalization and Denormalization: Functional dependencies, multivalued dependencies, normalization up to BCNF, 4NF, and 5NF. Denormalization concepts, need of denormalization, advantages, disadvantages, and trade-offs.

Indexing and Database Tuning: Primary indexes, secondary indexes, clus-tered and non-clustered indexes, B-tree and B+ tree indexes, hash indexes, query execution plans, query optimization, performance tuning, and basic cost-based optimization concepts.

UNIT IV: Distributed Databases, Data Warehousing, Security, Backup, and Cloud Databases 10 Hours

Distributed Databases: Distributed database concepts, fragmentation, replication, distributed query processing, distributed transaction management, consistency, availability, and basic CAP theorem awareness.

Data Warehousing and Mining Overview: Data warehouse architecture, ETL process, ELT awareness, star schema, snowflake schema, fact tables, dimension tables, OLAP basics, classification, clustering, and association-rule mining.

Database Security and Authorization: Database security models, users and roles, privileges, authentication, authorization, SQL injection, encryption, auditing, data privacy, and access-control mechanisms.

Backup, Recovery, and Cloud Databases: Backup types, recovery techniques, log-based recovery, checkpoints, cloud databases, managed database services, database scalability, availability, and disaster recovery.

Transactional Mode: Lecture method, e-team teaching, video-based learning, demonstration, peer discussion, open talk, cooperative teaching, flipped teaching, collaborative learning, database case studies, and hands-on SQL-based problem solving.

Suggested Readings:

- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts*. McGraw-Hill. https://books.google.com/books?isbn=9780078022159
- Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems*. Pearson. https://books.google.com/books?isbn=9780133970777
- Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems*. McGraw-Hill. https://books.google.com/books?isbn=9780072465631
- Date, C. J. (2003). *An Introduction to Database Systems*. Pearson. https://books.google.com/books?isbn=9780321197849
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book*. Pearson. https://books.google.com/books?isbn=9780131873254

Course Title:Object Oriented Programming using C++ Lab	**L**	**T**	**P**	**Cr.**
Course Code: MCA1402	0	0	2	1

Total Hours: 30

After completion of this course, the learner will be able to:

CO1. Explain the fundamentals of C++ structures, classes, objects, and basic procedural inputs/outputs.

CO2. Implement constructors, destructors, and operator overloading to manage object lifecycle and compile-time polymorphism.

CO3. Design and implement programs demonstrating inheritance, virtual functions, abstract classes, and run-time polymorphism.

CO4. Write C++ programs implementing templates, stream operations, file systems, and exception handling blocks.

LIST OF EXPERIMENTS

1. **Basic Class & Object**: Write a Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.
2. **Inline and Friend Functions**: Write a program to demonstrate the use of inline, friend functions, and `this` pointer.
3. **Static Members**: Write a program to implement static data members and static member functions.
4. **Object Lifecycle**: Write a Program to implement Default Constructor, Parameterized Constructor, and Destructor.
5. **Constructor Overloading**: Write a Program to demonstrate Constructor Overloading.
6. **Safe Copy Constructor**: Write a Program to demonstrate copy constructor for safe object copying (deep copy and memory ownership replication) instead of math calculations.
7. **Pointers and Memory**: Write a Program to allocate & deallocate memory using `new[]` and `delete[]`.
8. **Array of Pointers**: Write a Program to create an array of pointers.
9. **Polymorphism**: Write a Program to demonstrate the use of function overloading and swap two Characters of different data types.
10. **Operator Overloading**: Write a Program to overload comparison operator `operator==` and `operator!=`.
11. **Single Inheritance**: Create a base class containing the data member roll number and name. Also create a member function to read and display the

data. Create a derived class that contains marks of two subjects and total marks as the data members.

12. **Multilevel Inheritance**: Write a Program to create multilevel inheritance. (Classes A1, A2, A3).
13. **Function Overriding**: Write a program to demonstrate the concept of function overriding.
14. **Virtual Functions**: Write a Program to demonstrate the use of virtual functions and dynamic polymorphism.
15. **Abstract Classes**: Write a Program to demonstrate the concepts of pure virtual functions and abstract classes.
16. **Exception Handling**: Write a Program to perform exception handling. Use try-catch blocks to handle built-in and user-defined exceptions.
17. **File Copy Operations**: Write a Program to copy the contents of one file to another file.
18. **File I/O Streams**: Write a Program to Perform file operations such as read/write using streams (`ifstream`, `ofstream`, `fstream`).
19. **Templates & STL**: Write a Program to create Generic Functions using Template, and explore Standard Template Library (STL) containers (`vector`, `list`, `map`).

Transactional Mode: Demonstration, peer discussion, flipped teaching, collaborative learning, hands-on programming practice, and object-oriented lab exercises.

Suggested Readings:

- Balagurusamy, E. *Object-Oriented Programming with C++*. McGraw-Hill. https://books.google.com/books?isbn=9781259029936
- Josuttis, N. M. (2012). *The C++ Standard Library: A Tutorial and Reference*. Addison-Wesley. https://books.google.com/books?isbn=9780321623218
- C++ Reference Documentation. https://en.cppreference.com/w/
- Stroustrup, B. *The C++ Programming Language*. Addison-Wesley. https://books.google.com/books?isbn=9780321563842
- Learn C++ Tutorials. https://www.learncpp.com/

Course Title: Advanced Database Management System lab	**L**	**T**	**P**	**Cr.**
Course Code: MCA1403	0	0	2	1

Total Hours: 30

After completion of this course, the learner will be able to:

CO1. Design normalized database schemas utilizing ER models and integrity constraints.

CO2. Implement access control security policies through database administration (DBA) tools.

CO3. Write and troubleshoot advanced SQL queries, stored procedures, triggers, and cursors.

CO4. Implement concurrency control, query optimization, and star schema structures for data warehousing.

LIST OF EXPERIMENTS

1. **Schema Design**: Design a database schema for a university management system, including tables for students, courses, instructors, and enroll-ments, ensuring normalization and integrity constraints.
2. **Advanced Querying**: Write an SQL query to retrieve the names of students who have scored above the average marks in a particular subject using joins and subqueries.
3. **DBA Security Operations**: Implement database administration (DBA) tasks including user creation, roles configuration, privileges management, `GRANT`, and `REVOKE` operations.
4. **Stored Procedures**: Create a stored procedure to calculate the total salary of employees in a department and update the department's budget accord-ingly.
5. **Triggers**: Develop a trigger that automatically updates a log table whenever a new record is inserted into the main table, capturing details like timestamp and user ID.
6. **Transactions**: Write PL/SQL code to transfer funds from one bank account to another, ensuring that the transaction is atomic and consistent.
7. **Locking & Concurrency**: Implement a locking mechanism to prevent two users from booking the same hotel room simultaneously in a hotel reservation system.
8. **Performance Indexing**: Experiment with creating indexes on different columns of a large table and compare query performance using execution plans.
9. **Query Optimization**: Identify and optimize a slow-performing query by rewriting it or adding appropriate indexes to improve execution time.

10. **Data Warehousing**: Design a star schema for a retail sales data warehouse, including dimensions like time, product, and customer, and a fact table for sales transactions.
11. **OLAP Cube**: Implement an OLAP cube to analyze sales data, allowing users to slice and dice sales metrics by product category, region, and time period.
12. **Association Data Mining**: Implement the Apriori algorithm (corrected from "Priority algorithm") to discover frequent item sets from a transaction database and generate association rules.
13. **XML Storage**: Store and retrieve XML documents in a database, demonstrating XPath queries to extract specific information from XML data.
14. **Database Applications**: Develop small-scale database applications that integrate advanced DBMS concepts.

Transactional Mode: Demonstration, cooperative teaching, peer discussion, laboratory project works, and hands-on SQL practice.

Suggested Readings:

- Silberschatz, A. *Database System Concepts*. McGraw-Hill. https://books.google.com/books?isbn=9780078022159
- Feuerstein, S., & Pribyl, B. (2014). *Oracle PL/SQL Programming*. O'Reilly. https://books.google.com/books?isbn=9781449335021
- PostgreSQL Project Official Documentation. https://www.postgresql.org/docs/
- MySQL Reference Manual. https://dev.mysql.com/doc/refman/8.0/en/
- Casteel, J. (2015). *Oracle 12c: SQL*. Cengage. https://books.google.com/books?isbn=9781305251038

Course Title: Advanced Computer Architecture	**L**	**T**	**P**	**Cr.**
Course Code: MCA1404	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Explain the evolution of computer architectures, parallel programming models, and performance laws.

CO2. Analyze memory hierarchy designs, cache organization, shared memory spaces, and coherence.

CO3. Design and compare instruction-level processors, pipelined systems, and vector architectures.

CO4. Evaluate multiprocessor interconnect architectures, network-on-chip, and modern GPU/SIMT processor paradigms.

UNIT I: Parallel Computer Models and Performance Laws 15 Hours

Evolution of Computer architecture, system attributes to performance, Multiprocessors and multicomputers, Multi-vector and SIMD computers, PRAM and VLSI models. Parallelism in Programming, conditions for Parallelism, Program Partitioning and Scheduling, program flow Mechanisms. Speed up performance laws: Amdahl's law, Gustafson's law, Memory bounded speedup Model.

UNIT II: Memory Hierarchy, Caches, and Bus Systems 15 Hours

Memory hierarchy, cache and shared memory concepts, Cache memory organization, cache addressing models, Aliasing problem in cache, cache memory mapping techniques. Shared memory organization, Interleaved memory organization, Lower order interleaving, Higher order interleaving. Backplane bus systems: Bus addressing, arbitration and transaction. Bus cache coherence and consistency protocols.

UNIT III: Processor Pipelining and Instruction-Level Parallelism 15 Hours

Processor microarchitecture: CISC and RISC scalar processors, Superscalar processors, VLIW architecture. Pipelined Processors: Linear pipeline, nonlinear pipeline. Instruction pipeline design, Arithmetic pipeline design. Branch prediction, dynamic scheduling, out-of-order execution, vector processing principles, and GPU/SIMT execution awareness.

UNIT IV: Multiprocessors, Interconnection Networks, and Modern Trends 15 Hours

Multiprocessor system interconnects: Cross bar switch, Multiport Memory, Hot spot problem, Message passing mechanisms. Data flow computer architectures, Static and Dynamic dataflow. VLSI Computing Structures: Sys-

tolic array architecture, mapping algorithms into systolic arrays, Reconfigurable processor array. Open ISAs (RISC-V), network-on-chip (NoC), energy-performance trade-offs, and multi-core scalability.

Transactional Mode: Lecture, flipped teaching, peer discussion, collaborative learning, and case studies on modern processors.

Suggested Readings:

- Hennessy, J. L., & Patterson, D. A. (2017). *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann. https://books.google.com/books?isbn=9780128119051
- Hwang, K. (2016). *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. McGraw-Hill. https://books.google.com/books?isbn=9781259064647
- Stallings, W. (2015). *Computer Organization and Architecture*. Pearson. https://books.google.com/books?isbn=9780134101613
- Patterson, D. A., & Hennessy, J. L. (2013). *Computer Organization and Design*. Morgan Kaufmann. https://books.google.com/books?isbn=9780124077263
- Baer, J. L. (2010). *Microprocessor Architecture: From Simple Pipelines to Chip Multiprocessors*. Cambridge. https://books.google.com/books?isbn=9780521769921

Course Title: Software Project Management	**L**	**T**	**P**	**Cr.**
Course Code: MCA1405	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Define project scope, feasibility metrics, and select appropriate process lifecycle models.

CO2. Estimate project effort, costs, and timeline scheduling using WBS, COCOMO-II, and Gantt charts.

CO3. Analyze risk factors, implement project monitoring structures, change control, and quality assurance.

CO4. Manage project contracts, team behaviors, organizational structures, agile frameworks, and DevOps practices.

UNIT I: Project Definition and Process Model Selection 15 Hours

Project Definition, Activities Covered by Software Project Management, Overview of Project Planning, plan methods, methodology. Project Evaluation: Strategic Assessment, Technical Assessment, Cost Benefit Analysis, Cash Flow Forecasting, Cost Benefit Evaluation Techniques, selection of project approach, choice of process models, and project lifecycle constraints.

UNIT II: Software Estimation and Activity Scheduling 15 Hours

Work Breakdown Structure (WBS). Effort estimation: basics of software estimation, techniques, COCOMO-II, function point estimation, cost estimation, staffing pattern. Activity Planning: Objectives, Project Schedule, Sequencing and Scheduling Activities, Network Planning Models, Forward Pass, Back-ward Pass, Activity Float, Shortening Project Duration, Arrow Networks, and resource allocation.

UNIT III: Project Monitoring, Control, and Risk Management 15 Hours

Monitoring and Control: Creating Framework, Collecting the Data, Visualizing Progress, Cost Monitoring, Earned Value analysis, Prioritizing Monitoring, Getting Project Back to Target, and Change Control. Risk Management: Nature of Risk, Types of Risk, Managing Risk, Hazard Identification, Hazard Analysis, Risk Planning and Control, defect tracking, and project closure.

UNIT IV: Contract, Team, and Modern Project Governance 15 Hours

Managing Contracts: Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management, Acceptance. Managing People and Organizing Teams: Understanding Behavior, Selecting The Right Person, Motivation, Job Characteristics Model, Working In Groups,

Decision Making, Leadership, Organizational Structures. Stakeholder management. Agile project management, Scrum, Kanban, and CI/CD/DevOps awareness.

Transactional Mode: Lecture Method, peer discussion, flipped teaching, and collaborative software project case studies.

Suggested Readings:

- Hughes, B., & Cotterell, M. (2009). *Software Project Management*. McGraw-Hill. https://books.google.com/books?isbn=9780077122799
- Walker, R. (1998). *Software Project Management: A Unifted Framework*. Addison-Wesley. https://books.google.com/books?isbn=9780201309584
- Jalote, P. (2002). *Software Project Management in Practice*. Addison-Wesley. https://books.google.com/books?isbn=9780201737218
- Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge*. PMI. https://books.google.com/books?isbn=9781628256642
- Stellman, A., & Greene, J. (2005). *Applied Software Project Management*. O'Reilly. https://books.google.com/books?isbn=9780596009489

Course Title: Indian Knowledge System	**L**	**T**	**P**	**Cr.**
Course Code: IKS0016	4	0	0	4

Total Hours: 60

Course Learning Outcomes:

After completion of this course, the learner will be able to:

CO1. Develop a sense of rootedness and pride in India, along with an appreciation for its rich, diverse, ancient, and modern culture, knowledge systems, and traditions.

CO2. Understand the country's rich scientific and technological heritage.

CO3. Appreciate and promote the Indian concept of multidisciplinary learning systems, integrating them with modern science.

CO4. Recognize the importance of intellectual property rights in safeguarding Indian knowledge.

UNIT I **15 Hours**

Ancient philosophy of Knowledge: Vedas, Vedangas, sutras – Gurukul parampara

UNIT II **15 Hours**

Astronomy in India: The Beginnings of Indian Astronomy - The Early Historical Period-The Siddhāntic Era - The Kerala School - Aryabhatta - Varahamihira-Bhaskara I – Brahamagupta- Bhaskara II – Brief notes on Astronomical instruments

UNIT III **15 Hours**

Chemistry in India: Early Chemical Techniques, Atomism in Vaiśeṣika - Rishi Kanad- Nagarjuna- Al-Bīrūnī', Vāgbhaṭa- Sushruta- Carak Metallurgy in India - Definition, Metallurgy in Harappan Civilization, Metallurgy of Gold- Copper-Zinc- Bronze - Iron and steel.

UNIT IV **15 Hours**

Developments in Mathematics: Number systems- Geometry- works of Pingala-Baudhayana- Jaina Mahavira-Sridharacharya – Madhava Siddanthas and Calender systems

Transactional Mode: Seminars, Group discussion, Team teaching, Focused group discussion, Assignments, Project-based learning, Simulations, reflection and Self-assessment

Suggested Readings:

- Kapoor, K., & Singh, A. K. (2005). *Indian Knowledge Systems*. D.K. Printworld. https://books.google.com/books?isbn=9788124603369

- Mahadevan, B., Bhat, V. R., & Nagendra, N. (2022). *Introduction to Indian Knowledge System: Concepts and Applications*. PHI. https://books.google.com/books?isbn=9789391812102
- Alvares, C. (2002). *Decolonising History: Technology and Culture in India, China and the West*. Other India Press. https://books.google.com/books?isbn=9788185569567
- Joseph, G. G. (2009). *The Crest of the Peacock: Non-European Roots of Mathematics*. Princeton. https://books.google.com/books?isbn=9780691135267
- IKS Division Ministry of Education Project. https://iksindia.org/

Course Title: Cloud Computing	**L**	**T**	**P**	**Cr.**
Course Code: MCA1406	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Explain cloud architectures, delivery models (SaaS, PaaS, IaaS), and deployments.

CO2. Design virtualized infrastructures using hypervisors, Docker containers, and Kubernetes orchestration.

CO3. Evaluate cloud migration strategies, Service Level Agreements (SLAs), and financial operations (FinOps).

CO4. Analyze cloud security models, IAM, encryption, serverless computing, microservices, and Infrastructure as Code.

UNIT I: Cloud Fundamentals and Architecture **15 Hours**

Introduction to Cloud Computing: History and Evolution of Cloud Computing, Cloud Characteristics, Types of Cloud Computing, Cloud Computing Architecture, Basics of Cloud Infrastructure. Cloud Delivery Models: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), service combinations, and practical cloud applications.

UNIT II: Virtualization, Containers, and Infrastructure **15 Hours**

Virtualization Technologies: Understanding Hypervisors, Virtual machine management, CPU and I/O virtualization. Containerization: Introduction to Docker, container lifecycles, workload orchestration with Kubernetes. Cloud storage architectures, load balancing, auto-scaling, and software-defined networking basics.

UNIT III: Migration, SLAs, and Provider Selection **15 Hours**

Migrating to the Cloud: Broad Approaches to Migrating, Seven-step Model of Migration, Service Level Agreements (SLAs). Evaluating Business Need, Cloud vs. Hosted Applications, Cloud vs. Licensed Software. Selection of Cloud Provider: Business considerations, Interoperability, Portability and Integration, vendor lock-in, compliance, cost management, and FinOps awareness.

UNIT IV: Cloud Security, Serverless, and DevOps Trends **15 Hours**

Securing the Cloud: IAM, shared responsibility model, data encryption, compliance, logging, and disaster recovery. Serverless Computing: Function as a Service (FaaS), microservices architectures, cloud automation, Infrastructure as Code (IaC), CI/CD, Edge/Fog Computing, and multi-cloud strategies.

Transactional Mode: Lecture Method, e-team teaching, video based learning, and case studies on cloud deployments (AWS, Azure, GCP).

Suggested Readings:

- Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). *Mastering Cloud Computing*. McGraw-Hill. https://books.google.com/books?isbn=9781259029950
- Hwang, K., Fox, G. C., & Dongarra, J. J. (2011). *Distributed and Cloud Computing*. Morgan Kaufmann. https://books.google.com/books?isbn=9780123858801
- Erl, T., Cope, R., & Naserour, M. (2015). *Cloud Computing Design Patterns*. Prentice Hall. https://books.google.com/books?isbn=9780133858563
- Velte, A. T., Velte, T. J., & Elsenpeter, R. (2009). *Cloud Computing: A Practical Approach*. McGraw-Hill. https://books.google.com/books?isbn=9780071626941
- Marinescu, D. C. (2017). *Cloud Computing: Theory and Practice*. Morgan Kaufmann. https://books.google.com/books?isbn=9780128128107

Course Title: Big Data	**L**	**T**	**P**	**Cr.**
Course Code: MCA1407	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Discuss big data characteristics, pipelines, analytics lifecycle, and statistical foundations.

CO2. Implement distributed file systems (HDFS) and batch-processing using MapReduce execution.

CO3. Design real-time data pipelines using NoSQL databases, CAP theorem, Spark, and Kafka.

CO4. Analyze recommender algorithms, Graph link mining, and implement scalable analytics and governance.

UNIT I: Big Data Foundations and Analytics Lifecycle 15 Hours

Introduction to Big Data Platform: Challenges of Conventional Systems, Nature of Data, 6Vs of big data. Analytic Processes and Tools: Analysis Vs Reporting, Modern Data Analytic Tools, Statistical Concepts: Sampling Distributions, Re-Sampling, Statistical Inference, Prediction Error.

UNIT II: Distributed Storage and HDFS / MapReduce 15 Hours

Hadoop Environment: History of Hadoop, The Hadoop Distributed File System, Components of Hadoop, Scaling Out, Hadoop Streaming, Design of HDFS, Java interfaces to HDFS. MapReduce Framework: Developing a Map Reduce Application, How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Hadoop Cluster Setup, Configuration, and Security.

UNIT III: NoSQL Databases and Real-Time Stream Analytics 15 Hours

NoSQL Databases: Key-value, Document, Column-family, Graph databases, CAP theorem awareness. Mining Data Streams: Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, Decaying Window. Real Time Analytics Platform (RTAP) Applications: Spark and Kafka awareness, real-time pipelines.

UNIT IV: Link Analysis, Recommenders, and Visualization 15 Hours

Link Analysis: PageRank, Efficient Computation of PageRank, Topic-Sensitive Page Rank, Link Spam. Recommendation Systems: Content-Based Recommendations, Collaborative Filtering. Dimensionality Reduction. Visualizations: Visual data analysis techniques, dashboards. Data lakes, data governance, security, privacy, and scalable Machine Learning awareness.

Transactional Mode: Lecture, demonstration, peer discussion, flipped teaching, and big data case studies (Apache Hadoop, Spark, Kafka).

Suggested Readings:

- White, T. (2012). *Hadoop: The Deftnitive Guide*. O'Reilly. https://books.google.com/books?isbn=9781449311520
- Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). *Learning Spark*. O'Reilly. https://books.google.com/books?isbn=9781449358624
- Minelli, M., Chambers, M., & Dhiraj, A. (2013). *Big Data, Big Analytics*. Wiley. https://books.google.com/books?isbn=9781118147603
- Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of Massive Datasets*. Cambridge. https://books.google.com/books?isbn=9781108476348
- Apache Hadoop Project Official Documentation. https://hadoop.apache.org/docs/

Course Title: Machine Learning	**L**	**T**	**P**	**Cr.**
Course Code: MCA1408	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Explain machine learning paradigms, data preprocessing, feature representations, and Bayesian decision theory.

CO2. Implement linear discriminants, neural networks, backpropagation, and optimization methods.

CO3. Design decision trees, instance-based classifiers (k-NN), and apply quantitative evaluation metrics.

CO4. Evaluate computational learning theory (VC dimension), clustering methods, association rules, ensembles, and explainable AI (XAI).

UNIT I: ML Foundations, Data Preprocessing, and Bayes Decision Theory 15 Hours

Machine intelligence and applications, pattern recognition concepts: classification, regression, feature selection. Data preprocessing, feature representation, train-test-validation concepts, overfitting, underfitting, bias-variance trade-off. Bayesian decision theory: class conditional probability distributions, Bayes optimal classifier, learning classification approaches.

UNIT II: Linear Models, Gradient Descent, and Neural Networks 15 Hours

Linear machines: General and linear discriminants, decision regions, single layer neural network, linear separability, general gradient descent, perceptron learning algorithm, mean square criterion, and Widrow-Hoff learning algorithm (corrected from "Windrow-Hoff"). Multi-Layer Perceptron: two-layers universal approximators, back propagation learning, online and offline error surface, learning parameters, activation functions, and regularization.

UNIT III: Decision Trees, Instance-Based Learning, and Model Evaluation 15 Hours

Learning decision trees: Inference model, symbolic decision trees, entropy, mutual information, ID3 algorithm, C4.5 algorithm, continuous test nodes, confidence, pruning, learning with incomplete data. Instance-based Learning: Nearest neighbor classification, k-nearest neighbor. Model Evaluation: confusion matrix, precision, recall, F1-score, cross-validation, and regression metrics.

UNIT IV: Learning Theory, Clustering, and Explainable AI 15 Hours

Learning theory: formal model of the learnable, sample complexity, VC-dimension, inductive bias, Occam's razor, empirical risk, limitations. Unsupervised Learning: Clustering (K-Means, Hierarchical Clustering, DBSCAN). Association Rule

Learning: Apriori, Eclat. Ensembles. Explainable AI (XAI), fairness, privacy, and responsible ML awareness.

Transactional Mode: Lecture Method, peer discussion, flipped teaching, collaborative learning, and hands-on ML problem solving.

Suggested Readings:

- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. https://books.google.com/books?isbn=9780070428072
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. https://books.google.com/books?isbn=9780387310732
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. https://books.google.com/books?isbn=9780387848570
- Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT Press. https://books.google.com/books?isbn=9780262043793
- Scikit-Learn Official Library Documentation. https://scikit-learn.org/stable/

2 SEMESTER-II SYLLABUS

Course Title: Advanced Data Structure & Algorithms	L	T	P	Cr.
Course Code: MCA2450	3	0	0	3

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Analyze algorithm complexities, asymptotic notations, and design techniques (divide & conquer, greedy, DP).

CO2. Implement linear data structures (lists, stacks, queues) and select priority structures.

CO3. Design hashing mechanisms, separate chaining using linked list, and balanced trees (BST, AVL).

CO4. Implement advanced trees (B-trees, B+ trees), graph traversals, topological sorting, shortest paths, and spanning trees.

UNIT I: Complexity Analysis and Core Algorithm Design 15 Hours

Introduction to Data structures & Algorithms, Abstract Data Types. Performance Analysis: Space Complexity, Time Complexity, Asymptotic Notations (Big O, Omega, Theta), Performance measurement. Divide and Conquer, Greedy Method, Back Tracking Method, Dynamic programming. Searching and sorting algorithms: Bubble, Insertion, Radix, Quick, Merge, Heap, Selection, and Shell Sort. Linear/Binary Search.

UNIT II: Linear Data Structures, Stacks, and Queues 10 Hours

Linked lists: Singly, Doubly, and Circular Linked lists (Insert, delete, traverse, search). Stacks: Definition, Operations, Array/Linked List implementations, Applications (Postfix evaluation, parenthesis balancing). Queues: Definition, Operations, simple queue. Types of queues: Circular, Double ended, Priority queue (Implementation using linked list).

UNIT III: Hashing, Priority Queues, and Search Trees 10 Hours

Hashing: Hashing Techniques, Address calculation Techniques, common hash functions, Collision resolution: Linear probe, Quadratic probe, Key offset, Rehashing, Double hashing, separate chaining using linked list (corrected from "Link list addressing"). BST: Definition, Operations. AVL tree: Definition, rotations. Heaps: Definition, Operations.

UNIT IV: Advanced Trees, Graph Algorithms, and Modern Interconnects 10 Hours

Advanced Trees: Huffman tree, Expression tree, threaded binary tree (cor-rected from "Binary threaded three"), M-way Tree, B-tree, B+ tree awareness.

Graphs: Representations (Adjacency Matrix/List), traversals (DFS, BFS), Topological Sorting. Shortest Path: Dijkstra's, Bellman-Ford, Floyd-Warshall. MST: Prim's and Kruskal's. Union-find awareness.

Transactional Mode: Lecture Method, peer discussion, flipped teaching, and collaborative implementation exercises.

Suggested Readings:

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms*. MIT Press. https://books.google.com/books?isbn=9780262046305
- Kleinberg, J., & Tardos, E. (2006). *Algorithm Design*. Pearson. https://books.google.com/books?isbn=9780321295354
- Sedgewick, R., & Wayne, K. (2011). *Algorithms*. Addison-Wesley. https://books.google.com/books?isbn=9780321573513
- Horowitz, E., Sahni, S., & Silicon Press. *Fundamentals of Computer Algorithms*. Silicon Press. https://books.google.com/books?isbn=9780929306414
- Knuth, D. E. (2011). *The Art of Computer Programming*. Addison-Wesley. https://books.google.com/books?isbn=9780321751041

Course Title: Programming Using Python	**L**	**T**	**P**	**Cr.**
Course Code: MCA2451	3	0	0	3

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Explain Python variables, datatypes, expressions, and input/output print mechanisms.

CO2. Apply conditional blocks, loops, recursive functions, modules, and random generators.

CO3. Analyze Python core data structures (strings, lists, tuples, sets, dictionaries) and regex.

CO4. Implement file handling, OOP classes, SQLite interfaces, and basic web framework architectures.

UNIT I: Python Basics, Variables, and Expressions 15 Hours

Introduction to Python 3 (Python 3.11/3.12 recommended), interpreted high-level nature, interactive and script modes. Variables, Expressions, and Statements: Values, Variables, keywords. Operators: Arithmetic, relational, logi-cal, precedence. Input/Output: input(), print(), f-strings formatting, putting comments.

UNIT II: Control Flow, Functions, and Modules 10 Hours

Conditional constructs: if-else statement. Loops: while, for, range(), break, continue, pass. Nested loops. Functions: defining, invoking, arguments, parameters, scope, recursion, returning values. Modules: importing, math module, random/randint generators, creating user-defined modules.

UNIT III: Core Python Data Structures and Regex 10 Hours

Strings: creating, indexing, operators, functions (len, find, split, replace, upper, lower, etc.). Lists: mutable lists, traversals, slices, insertions, deletions, list methods (insert, append, sort, reverse). Tuples: immutable concept, tuple functions. Dictionaries: key-value pairs, traversals, updates, methods (clear, get, items, keys, values). Sets and comprehensions. Regular Expressions (regex) validation.

UNIT IV: File Handling, Exceptions, OOP, and DB Connections 10 Hours

File Handling: reading/writing text/CSV files, with open(). Exception Handling: try-except-finally blocks, custom exceptions. Object-Oriented Programming (OOP): classes, objects, inheritance, methods. Database Connectivity: SQLite/MySQL CRUD. Web frameworks: Django/Flask awareness, unit testing, debugging, docstrings.

Transactional Mode: Lecture, demonstration, peer discussion, collaborative coding, and case studies on web application integrations.

Suggested Readings:

- Lutz, M. (2013). *Learning Python*. O'Reilly. https://books.google.com/books?isbn=9781449355739
- Matthes, E. (2019). *Python Crash Course*. No Starch Press. https://books.google.com/books?isbn=9781593279288
- Sweigart, A. (2019). *Automate the Boring Stuff with Python*. No Starch Press. https://books.google.com/books?isbn=9781593279929
- Beazley, D., & Jones, B. K. (2013). *Python Cookbook*. O'Reilly. https://books.google.com/books?isbn=9781449340377
- Python Language Official Documentation. https://docs.python.org/3/

Course Title: Advanced Data Structures & Lab	**L**	**T**	**P**	**Cr.**
Course Code: MCA2452	0	0	2	1

Total Hours: 30

After completion of this course, the learner will be able to:

CO1. Implement static memory structures (arrays, matrices, sparse matrices) and basic sorting/searching.

CO2. Write recursive functions, allocate memory dynamically, and implement linked list structures.

CO3. Design and implement stack, queue, circular queue, and hashing structures.

CO4. Develop AVL, BST, heap structures, and graph algorithms (DFS, BFS, shortest path, mini-project).

LIST OF EXPERIMENTS

1. **Arrays Block**: Write programs for 1-D arrays insertion, deletion, and reversal without using another variable.
2. **Matrices**: Write programs for 2-D matrices, matrix transpose, matrix multiplication, and sparse matrix representations.
3. **Searching & Sorting**: Implement linear search, binary search, bubble sort, selection sort, and insertion sort.
4. **Recursion & Allocation**: Write programs to find factorial and print Fibonacci series using recursion, and perform dynamic memory allocation.
5. **Linked Lists**: Write programs to input, traverse, and reverse a singly linked list, and perform insertions/deletions.
6. **Linked List Variants**: Write programs for doubly linked lists and circular linked lists.
7. **Stack Operations**: Write programs to implement stack operations (Array and Linked list) and applications (postfix evaluation, parenthesis valida-tion).
8. **Queue Operations**: Write programs to implement simple queues, circular queues, and priority queues.
9. **Hashing**: Implement hashing functions and linear/quadratic probing and separate chaining.
10. **BST Trees**: Write programs to input, insert, and delete in a Binary Search Tree (BST).
11. **AVL Trees**: Implement AVL trees, including merged LL, RR, LR, RL rotation algorithms.

12. **Heaps**: Write programs to implement min-heaps, max-heaps, and heap sort.
13. **Graphs**: Write programs to input a graph, print adjacency list, and perform DFS and BFS traversals.
14. **Shortest Path**: Implement Dijkstra's shortest path method.
15. **File Optimization Project**: Implement file-based sorting/searching, duplicate deletion in arrays, and a small-scale mini-project combining multiple structures.

Transactional Mode: Demonstration, peer discussion, flipped teaching, collaborative learning, hands-on programming practice.

Suggested Readings:

- Cormen, T. H. *Introduction to Algorithms*. MIT Press. https://books.google.com/books?isbn=9780262046305
- Weiss, M. A. (2013). *Data Structures and Algorithm Analysis in C++*. Pearson. https://books.google.com/books?isbn=9780132847377
- GeeksforGeeks Data Structures & Algorithms. https://www.geeksforgeeks.org/data-structures/
- LeetCode Coding Platform. https://leetcode.com/
- HackerRank Algorithms Track. https://www.hackerrank.com/domains/algorithms

Course Title: Python Lab	**L**	**T**	**P**	**Cr.**
Course Code: MCA2453	0	0	2	1

Total Hours: 30

After completion of this course, the learner will be able to:

CO1. Write basic Python console programs utilizing datatypes, f-strings, and math libraries.

CO2. Apply loops, conditional checking, and modular functions.

CO3. Manipulate compound data elements (lists, dicts, tuples, sets) and text regexes.

CO4. Implement file writing/reading, exception try-excepts, and database CRUD mini-projects.

LIST OF EXPERIMENTS

1. **Basics**: Program to print hello world, add numbers, concatenate strings, and take input from user.
2. **Control Flow**: Program to demonstrate conditional checking (if-else), loops (for, while), and loop control (break, continue, pass).
3. **Functions**: Programs demonstrating user-defined functions, scope, math library imports, random/randint generators, and command-line arguments.
4. **Lists**: Program to demonstrate list operations, insertions, reversals, sorting, and filtering even numbers.
5. **Tuples & Sets**: Write programs to demonstrate tuple packing/unpacking and set operations.
6. **Dictionaries**: Write programs for dictionary keys, values, and updates, and count word frequency in a list.
7. **Exceptions**: Implement exception handling using try-except-finally blocks and raise custom exceptions.
8. **File & RegEx**: Write programs for reading/writing text and CSV files, count lines/words, and validate email/phone format using regular expressions.
9. **OOP**: Write a program demonstrating class creation, constructors, inheritance, and method overriding.
10. **Database integration**: Write a program to connect to an SQLite database and perform CRUD (create, read, update, delete) operations.
11. **Mini-Project**: Build a console database application (e.g. library registry or student logs) to integrate the above concepts.

Transactional Mode: Demonstration, peer discussion, flipped teaching, collaborative learning, hands-on programming practice.

Suggested Readings:

- Lutz, M. *Learning Python*. O'Reilly. https://books.google.com/books?isbn=9781449355739
- McKinney, W. (2017). *Python for Data Analysis*. O'Reilly. https://books.google.com/books?isbn=9781491957660
- Real Python Tutorials. https://realpython.com/
- W3Schools Python Reference. https://www.w3schools.com/python/
- Project Euler Programming Challenges. https://projecteuler.net/

Course Title: Advanced Operating Systems	**L**	**T**	**P**	**Cr.**
Course Code: MCA2454	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Explain kernel/user modes, system call interfaces, and process/thread models.

CO2. Analyze CPU scheduling algorithms, IPC synchronization (semaphores, monitors), and deadlock handling (avoidance, prevention).

CO3. Design memory allocation, paging, file systems, disk space allocation, and buffering systems.

CO4. Evaluate Linux kernel models, distributed OS, security mechanisms, virtualization hypervisors, and Docker/Kubernetes container orchestration.

UNIT I: Kernel Architecture and Process Thread Models 15 Hours

Operating system functions and characteristics, Operating System classification, historical evolution, Real time system, Distributed system. Methodologies for implementation of O/S service, system calls, system programs, Interrupt mechanisms. Processes: Process Concept, Process model, Process scheduling, Process states, hierarchies, Context switching, Process synchronization. Threads: thread concepts, multi-threading models, and process tables.

UNIT II: IPC, Concurrency Bottlenecks, and Deadlock Handling 15 Hours

Inter-process communication: Race conditions, critical sections, mutual exclusion, Peterson's solution, producer-consumer problem (corrected from "producer-customer"), Reader-Writer's, Dining Philosophers. Semaphores, mutex, monitors, message passing. Process scheduling: criteria, scheduling algorithms (Round robin, Priority, FCFS, SJF, Multilevel feedback). Deadlocks: Characteristics, modeling, detection, recovery, avoidance, prevention, and deadlock handling (corrected from "deadlock hand").

UNIT III: Memory Allocation, Page Management, and File Systems 15 Hours

Memory Management: Allocation, Virtual Memory, Logical vs Physical address space, Multiprogramming with fixed/variable partitions, paging, demand paging, page tables, inverted page tables, page replacement algorithms, page fault handling, working set model, local vs global allocation, page size, segmentation. File systems: File operations, directory structures, path names, implementation of file systems, Allocation methods (Contiguous, linked, index allocation, block allocation, free space management). Device management: buffering, caching, spooling, and disk scheduling.

UNIT IV: Linux Internals, Virtualization, and Cloud Containers 15 Hours

Linux kernel architecture: Process/memory/file management, Linux security model, hypervisors (VMware, KVM, Hyper-V). Cloud OS and containerization: Docker, Kubernetes, namespaces, cgroups, distributed transaction protocols, OS-level security (Authentication, Authorization, Access Control, Intrusion detection).

Transactional Mode: Lecture, flipped teaching, peer discussion, collaborative learning, and case studies on OS setups.

Suggested Readings:

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. Wiley. https://books.google.com/books?isbn=9781119456339
- Tanenbaum, A. S., & Bos, H. (2014). *Modern Operating Systems*. Pearson. https://books.google.com/books?isbn=9780133591620
- Stallings, W. (2017). *Operating Systems: Internals and Design Principles*. Pearson. https://books.google.com/books?isbn=9780134670959
- Love, R. (2010). *Linux Kernel Development*. Addison-Wesley. https://books.google.com/books?isbn=9780672329463
- Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books. https://books.google.com/books?isbn=9781985010505

Course Title: Advanced Computer Graphics	**L**	**T**	**P**	**Cr.**
Course Code: MCA2455	3	0	0	3

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Explain 2D/3D transformations, homogeneous coordinates, graphics pipelines, windowing, and clipping.

CO2. Design mathematical curves (Hermite, Bezier, B-spline), surfaces, and solid modeling (CSG, B-rep).

CO3. Analyze visibility culling algorithms, illumination models (ambient, diffuse, specular), and recursive ray tracing.

CO4. Evaluate modern graphics APIs (OpenGL, Vulkan), GLSL shader programming, physically-based rendering (PBR), and VR/AR animations.

UNIT I: Transformations, Viewing, and Graphics Pipelines 10 Hours

Review of two-dimensional graphics: Transformations, Windowing, Clipping. Three Dimensions: 3D geometry, primitives and transformations. Rotation about an arbitrary axis, Parallel and perspective projection. Viewing parameters, 3D clipping, viewing transformations, and graphics pipeline structure.

UNIT II: Curves, Splines, and Solid Geometric Modeling 10 Hours

Curves and Fractals: Polygon Meshes. Parametric Cubic curves: B-spline, Bezier, Hermite. Parametric Bicubic Surfaces, Quadric surfaces. Fractals: fractal lines and surfaces. Solid Modeling: Representing solids, Regularized Boolean Set Operations, Primitive Instancing, Sweep and Boundary Representations, Spatial-partitioning, Constructive Solid Geometry (CSG). Color Models: CIE chromaticity diagram, Color models for Raster Graphics (RGB, HSV, YUV).

UNIT III: Hidden Surfaces, Visibility, and Photorealistic Rendering 10 Hours

Hidden Lines and Surfaces: Algorithms for Visible-Line and Surface determination: z-buffer, List priority, Scan line, Area Subdivision, Ray Tracing. Illumination and Shading: Surface detail, shadows, transparency, texture mapping awareness, ambient, diffuse, specular illumination. Shading models: Flat, Gouraud, and Phong shading. Recursive Ray Tracing.

UNIT IV: GPU Programming Shaders, APIs, and VR / AR 15 Hours

Image based Rendering. Graphics APIs: OpenGL, DirectX, Vulkan (comparison), GPU Programming with GLSL or HLSL, programmable graphics pipeline. Real-Time Rendering and Game Engines, Physically Based Rendering (PBR), Non-Photorealistic Rendering (NPR). Animation Techniques: Keyframing, Skeletal Animation, VR and AR. Graphics in AI and Machine Learning.

Transactional Mode: Lecture, flipped teaching, peer discussion, cooperative teaching, collaborative learning.

Suggested Readings:

- Hearn, D., Baker, M. P., & Carithers, W. (2010). *Computer Graphics with OpenGL*. Pearson. https://books.google.com/books?isbn=9780136053583
- Foley, J. D., Van Dam, A., Feiner, S. K., & Hughes, J. F. (1995). *Computer Graphics: Principles and Practice*. Addison-Wesley. https://books.google.com/books?isbn=9780201848403
- Shirley, P., & Marschner, S. (2009). *Fundamentals of Computer Graphics*. A K Peters. https://books.google.com/books?isbn=9781568814698
- Angel, E., & Shreiner, D. (2011). *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*. Pearson. https://books.google.com/books?isbn=9780132545235
- OpenGL Official Khronos Specification. https://www.opengl.org/registry/

Course Title: ADVANCED COMPUTER GRAPHICS LAB	**L**	**T**	**P**	**Cr.**
Course Code: MCA2456	0	0	2	1

Total Hours: 30

After completion of this course, the learner will be able to:

CO1. Set up modern OpenGL/GLFW libraries and write basic primitive transformations.

CO2. Model curves (Bezier, Hermite) and surfaces with c0/c1 continuity.

CO3. Implement visibility algorithms (Z-buffer) and illumination shading models (ambient, diffuse, specular).

CO4. Write GLSL programmable shaders, particle simulators, and configure stereoscopic VR scenes.

LIST OF EXPERIMENTS

1. **Setup & Primitives**: Set up OpenGL 3.3+ (GLFW/FreeGLUT) and draw basic geometric primitives.
2. **Transformations**: Implement 2D and 3D coordinate transformations (translation, rotation, scaling).
3. **Curve Modeling**: Write a program to draw Bezier Curves.
4. **Continuity**: Implement Bezier curves with C0 and C1 continuity constraints.
5. **Hermite Curves**: Write a program to perform Hermite Curve modeling.
6. **Visibility Culling**: Draw a 3D Cube with and without back-face culling options.
7. **Hidden Surface**: Implement the Z-buffer visible surface algorithm.
8. **Illumination Models**: Write programs to implement Flat, Gouraud, and Phong shading models.
9. **Light Sources**: Implement ambient, diffuse, and specular light sources.
10. **Texture Mapping**: Apply texture mapping to a 3D object.
11. **GLSL Programmable Shaders**: Write simple vertex and fragment shaders using GLSL 330.
12. **Physically-Based Rendering**: Implement a basic PBR pipeline using shaders.
13. **Non-Photorealistic Rendering**: Create cartoon rendering (cell shading) or sketch effects.
14. **Particle Simulator**: Create a basic particle system to simulate fire, smoke, or water with motion and collision.
15. **Virtual Reality**: Implement stereoscopic rendering layout for VR.

16. **Graphics Mini-Project**: Build an interactive 3D graphics application (e.g. a simple game scene or visualizer).

Transactional Mode: Demonstration, peer discussion, flipped teaching, collaborative learning, hands-on programming practice.

Suggested Readings:

- Shreiner, D., Sellers, G., Kessenich, J., & Licea-Kane, B. (2013). *OpenGL Programming Guide*. Addison-Wesley. https://books.google.com/books?isbn=9780321773036
- WebGL Specification Khronos Alliance. https://registry.khronos.org/webgl/specs/latest/2.0/
- Learn OpenGL Shaders Tutorials. https://learnopengl.com/
- Angel, E. *Interactive Computer Graphics*. Pearson. https://books.google.com/books?isbn=9780132545235
- Three.js Javascript 3D Engine Library. https://threejs.org/

Course Title: ADVANCED WEB TECHNOLOGIES	**L**	**T**	**P**	**Cr.**
Course Code: MCA2457	0	0	4	2

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Design responsive, semantic frontends using HTML5, CSS3 Grid/Flexbox, and Bootstrap/Tailwind.

CO2. Write dynamic JavaScript DOM manipulations, forms validation, and calendar controls.

CO3. Parse structured data using XML (with DTD/XSD validations) and JSON.

CO4. Develop persistent full-stack Java web applications using JDBC, Servlets, JSP (with Tomcat 10+ and jakarta.servlet), session management, and MVC architectures.

LIST OF EXPERIMENTS

1. **Frontend Layouts**: Create a semantic HTML5 page using multimedia assets and design a responsive CSS3 grid/flexbox layout with media queries.
2. **Dynamic JavaScript DOM**: Design a dynamic registration webpage with JavaScript DOM validations.
3. **Client-side Logic**: Develop a JavaScript calculation page containing buttons for Factorial, Fibonacci, Prime, and Palindrome with DOM rendering.
4. **JS Objects**: Write JavaScript programs to utilize Math, String, Array, and Date objects.
5. **Event Handling**: Write JavaScript event handlers for mouse-over, window opening, background color changes, and dynamic calendar combo box rendering.
6. **Data Exchange**: Write XML documents validated against DTD and XSD schemas, and implement JSON parsing.
7. **JDBC Connection**: Connect a Java program to a DBMS (MySQL/SQLite) and perform full JDBC CRUD operations.
8. **Servlets Basics**: Develop a Java Servlet to display a welcome message and handle forms data.
9. **Servlet Authentication**: Develop a Servlet to validate user authentication against credentials stored in servlet config, and connect it with database check.
10. **Student Registry**: Write a Servlet registration program to store form values inside database tables.
11. **JSP Dynamic Content**: Develop JSP pages displaying dynamic contents, and write a database backed JSP CRUD application.

12. **Session & Cookies**: Implement state management inside Servlet/JSP using session attributes and cookies.
13. **AJAX / Fetch API**: Implement asynchronous client-server calls using AJAX or Fetch API.
14. **MVC Web Application**: Build an MVC architecture application integrating Servlet-JSP-JDBC CRUD.
15. **Web Mini-Project**: Build an integrated full-stack web application (e.g. student ledger or portal) using Tomcat 10+ (jakarta.servlet APIs) and version control.

Transactional Mode: Demonstration, peer discussion, flipped teaching, collaborative learning, hands-on programming practice.

Suggested Readings:

- Duckett, **J.** (2011). *HTML & CSS: Design and Build Web Sites*. Wiley. https://books.google.com/books?isbn=9781118008188
- Duckett, **J.** (2014). *JavaScript & JQuery: Interactive Front-End Web Development*. Wiley. https://books.google.com/books?isbn=9781118531648
- Haverbeke, M. (2018). *Eloquent JavaScript*. No Starch Press. https://books.google.com/books?isbn=9781593279509
- Mozilla Developer Network Web Docs. https://developer.mozilla.org/
- World Wide Web Consortium Standards. https://www.w3.org/TR/

Course Title: IoT & Its Applications	**L**	**T**	**P**	**Cr.**
Course Code: MCA2458	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Explain IoT architectures, enabling technologies, layered stacks, sensors, and actuators.

CO2. Program embedded hardware platforms (Arduino, Raspberry Pi, ESP32) with sensor inputs.

CO3. Compare access, network, and application protocols (Zigbee, LoRaWAN, 6LoWPAN, MQTT, CoAP) and secure IoT networks.

CO4. Implement IoT analytics, cloud integration (Firebase, Node-RED), Edge AI, and TinyML algorithms.

UNIT I: IoT Ecosystems, Sensors, and Actuators 15 Hours

IoT evolution, enabling technologies, M2M Communication, IoT World Forum (IoTWF) standardized architecture, Simplified IoT Architecture, Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem. Sensors, Actuators, Smart Objects, and connecting Smart Ob-jects.

UNIT II: Embedded Computing Hardware Platforms 15 Hours

Design Methodology. Embedded computing logic, Microcontrollers, System on Chips (SoC), GPIO, ADC, PWM, UART, SPI, I2C. Hardware platforms: Arduino boards, Raspberry Pi, ESP32/ESP8266 programming, sensor interface logic, Python 3 and Arduino IDE 2.x environments.

UNIT III: Communication Protocols and IoT Security 15 Hours

IoT Access Technologies: IEEE 802.15.4, Zigbee, BLE, Wi-Fi (802.11ah), and LoRaWAN (corrected from "Lora WAN"). Network Layer: IPv4/IPv6, 6LoWPAN, RPL. Application Layer: CoAP, MQTT (MQTT 5.0 awareness), HTTP/HTTPS, WebSockets. SCADA treated as industrial control/monitoring system. Ded-icated IoT security: threat models, device authentication, encryption, secure OTA updates, and privacy.

UNIT IV: Data Analytics, Cloud Platforms, and Edge AI Trends 15 Hours

IoT Data Analytics: Structured vs Unstructured Data, Data in Motion vs Data at Rest, Analytics challenges. Supporting Services: Cloud computing database integration (Firebase, ThingSpeak), Node-RED, Blynk, AWS/Azure IoT services. Applications: Smart Home, Smart Cities, IIoT (Industry 4.0), healthcare, agriculture, wearables. Future Trends: Edge AI, TinyML, Blockchain in IoT, 5G, digital twins, green IoT.

Transactional Mode: Lecture Method, peer discussion, flipped teaching, collaborative learning, and hands-on IoT prototyping.

Suggested Readings:

- Bahga, A., & Madisetti, V. (2014). *Internet of Things: A Hands-On Ap-proach*. VPT. https://books.google.com/books?isbn=9780996025515
- Hersent, O., Boswarthick, D., & Elloumi, O. (2012). *The Internet of Things: Key Applications and Protocols*. Wiley. https://books.google.com/books?isbn=9781119994350
- Minoli, D. (2013). *Building the Internet of Things with IPv6 and MIPv6*. Wiley. https://books.google.com/books?isbn=9781118473474
- IEEE Power and Energy Society Publications. https://www.ieee-pes.org/
- Pfister, C. (2011). *Getting Started with the Internet of Things*. O'Reilly. https://books.google.com/books?isbn=9781449393571

Course Title: Digital Image Processing	**L**	**T**	**P**	**Cr.**
Course Code: MCA2459	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Understand the fundamentals of digital image and its processing

CO2. Apply image enhancement techniques in spatial and frequency domain.

CO3. Elucidate the mathematical modeling of image restoration and compression

CO4. Describe object detection and recognition techniques.

UNIT I **15 Hours**

Background: Introduction to electronic systems for image transmission and storage, computer processing and recognition of pictorial data, overview of practical applications.

Fundamentals: Mathematical and perceptual preliminaries, human visual system model, image signal representation, imaging system specification building image quality, role of computers, image date formats.

UNIT II **15 Hours**

Image Processing Techniques: Image enhancement, image restoration, image feature extraction, image data compression and statistical pattern recognition.

Hardware Architecture for image processing: Distributed processing of image data, role of array processing, standard image processor chips (as example).

UNIT III **15 Hours**

Techniques of color image processing: Color image signal representation, color system transformations, extension of processing techniques to color do-main.

UNIT IV **15 Hours**

Applications of Image Processing: Picture data archival, machine vision, med-ical image processing. Image Segmentation: Edge detection: Sobel, Prewitt, Canny, Thresholding techniques: Global and adaptive, Region-based segmen-tation, Morphological operations: Dilation, erosion, opening, closing, Object recognition and classification, Introduction to deep learning in image analysis (CNN basics)

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing*. Pearson. https://books.google.com/books?isbn=9780133356724
- Pratt, W. K. (2007). *Digital Image Processing*. Wiley. https://books.google.com/books?isbn=9780471767770
- Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2009). *Digital Image Processing Using MATLAB*. Gatesmark Publishing. https://books.google.com/books?isbn=9780982085400
- OpenCV Library Developer Reference. https://docs.opencv.org/
- Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice Hall. https://books.google.com/books?isbn=9780133361650

Course Title: Data Warehousing and Data Mining Techniques	**L**	**T**	**P**	**Cr.**
Course Code: MCA2460	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Identify the scope and necessity of Data Mining & Warehousing for the society

CO2. Describe the designing of Data Warehousing so that it can be able to solve the root problems.

CO3. Remove redundancy and incomplete data from the dataset using data preprocessing methods.

CO4. Develop a data mining application for data analysis using various tools.

UNIT I **15 Hours**

Introduction: Data Warehousing: Definition, Characteristics of a Data Warehouse, Data warehouse Usage, DBMS vs. Data warehouse

Developing Data Warehouse: Data warehousing components, Steps and Crucial decisions for the design and construction of Data Warehouses, Three-tier Data warehouse architecture, Data Warehouse Implementation, Design, performance and technological considerations, Metadata.

UNIT II **15 Hours**

Developing Data Mart based Data Warehouse Types of data marts, Metadata for a data mart, Data model for a data mart, Maintenance of a data mart, Software components for a data mart, Performance issues, Security in data mart.

OLAP Systems Types of OLAP, Relational vs. Multidimensional OLAP, Data modeling: Star schema, Snowflake schema, OLAP tools

UNIT III **15 Hours**

Data Mining: Introduction to data mining, Data mining process, Major issues and Application of Data mining, Data preprocessing: Data cleaning, Data integration and transformation and Data reduction; Tools for data mining.

Data Mining Techniques: Association rules: Introduction, Market basket analysis, Frequent Pattern Mining algorithms: Apriori algorithm, Partition algorithm.

UNIT IV **15 Hours**

Classification and Prediction: Definition, Issues regarding Classification and Prediction, Classification by Decision Tree Induction, Support Vector Machines, k- Nearest-Neighbor, Prediction: Linear and Non-Linear Regression

Clustering: Definition, Types of data in cluster analysis, clustering paradigms: K-Means and K-Medoids, Mining Sequence patterns: Generalized Sequential Patterns (GSP) mining algorithm, Hidden Markov Model, Social Network Analysis. Advanced Topics and Applications Web mining: Web content, structure, and usage mining, Text mining and opinion mining basics, Temporal and spa-tial data mining, Big data analytics and Hadoop ecosystem overview, Appli-cations of data mining in business, healthcare, e-commerce, finance, Ethical issues and privacy in data mining

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Han, **J.,** Kamber, M., & Pei, **J.** (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann. https://books.google.com/books?isbn=9780123814791
- Tan, P. N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Pearson. https://books.google.com/books?isbn=9780321321367
- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit*. Wiley. https://books.google.com/books?isbn=9781118530801
- Dunham, M. H. (2003). *Data Mining: Introductory and Advanced Topics*. Pearson. https://books.google.com/books?isbn=9780130888921
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann. https://books.google.com/books?isbn=9780123748560

3 SEMESTER-III SYLLABUS

Course Title: Artificial Intelligence	L	T	P	Cr.
Course Code: MCA3500	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Differentiate the various searching techniques, constraint satisfaction problem.

CO2. Classify the role of agents and the way of evaluating it.

CO3. Analyze and design a real-world problem for implementation and understand the dynamic behavior of a system.

CO4. Compare different machine learning techniques to design AI machines and enveloping applications for real world problems.

UNIT I **15 Hours**

Introduction to Artificial Intelligence (AI) and Problem Space: Introduction AI technique, Turing test, History and developments in AI, applications of AI, State space representation, production systems, systematic control strategies: Breadth first search and Depth first search, problem characteristics, product system characteristics, issues in the design of search programs.

Heuristic Search Technologies: Introduction to heuristic search, Generate and test, Hill Climbing, Best First search, A*, Problem reduction, AO*.

UNIT II **15 Hours**

Knowledge representation methods - Propositional logic and first order predicate logic, Resolution principle, Semantic networks, partitioned semantic nets, Frames, Scripts and conceptual dependencies.

Game playing: Minimax search procedure, reducing alternatives using Alpha-Beta pruning method examples.

UNIT III **15 Hours**

Approaches to AI: Turing Test and Rational Agent Approaches; State Space Representation of Problems, Heuristic Search Techniques, Game Playing, Min-Max Search, Alpha Beta Cutoff Procedures.

Natural Language Processing: Grammar and Language; Parsing Techniques, Semantic Analysis and Pragmatics.

UNIT IV **15 Hours**

Tools and Technologies for AI: Introduction to AI language, Natural Language Processing (NLP), Ethical and Social Implications of AI, AI Trends and Future

Directions,

LISP: Symbolic expression, creating, appending and modifying lists, defining functions, Predicates, Conditionals, Recursion, Iteration, Printing and reading, Lambda expressions and higher order function, List storage.

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Russell, S., & Norvig, P. (2020). *Artiftcial Intelligence: A Modern Approach*. Pearson. https://books.google.com/books?isbn=9780134610993
- Rich, E., Knight, K., & Nair, S. B. (2009). *Artiftcial Intelligence*. McGraw-Hill. https://books.google.com/books?isbn=9780070087705
- Nilsson, N. J. (2014). *Principles of Artiftcial Intelligence*. Morgan Kauf-mann. https://books.google.com/books?isbn=9780934613118
- Luger, G. F. (2008). *Artiftcial Intelligence: Structures and Strategies for Complex Problem Solving*. Addison-Wesley. https://books.google.com/books?isbn=9780321545817
- Association for the Advancement of Artificial Intelligence. https://www.aaai.org/

Course Title: Research Methodology	**L**	**T**	**P**	**Cr.**
Course Code: MCA3501	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Understand key research methodology concepts and issues

CO2. Identify the role and importance of research in the Computer Applications

CO3. Identify the concepts and procedures of sampling, data collection, analysis and reporting.

CO4. Analyze appropriate research problem and parameters

CO5. Implement the basic concepts of research and its methodologies

UNIT I **15 Hours**

Research: its concept, nature, scope, need and Objectives of Research, Research types, Research methodology, Research process – Flow chart, descrip-tion of various steps, Selection of research problem.

UNIT II **15 Hours**

Research Design: Meaning, Objectives and Strategies of research, different research designs, important experimental designs,

Methods of Data Collection and Presentation: Types of data collection and classification, Observation method, Interview Method, Collection of data through Questionnaires, Schedules, data analysis and interpretation, editing, coding, content analysis and tabulation

UNIT III **15 Hours**

Sampling Methods: Different methods of Sampling: Probability Sampling methods, Random Sampling, Systematic Sampling, Stratified Sampling, Cluster Sampling and Multistage Sampling. Non probability Sampling methods, Sample size.

UNIT IV **15 Hours**

Report writing and Presentation: , Interpretation and Report Writing, Types of reports, Report Format/ Thesis – Cover page, Introductory page, Text, Bibliography, Appendices, Typing instructions, Oral Presentation, Thesis, Referencing Styles: APA, MLA, Chicago, Bibliography and Citation Tools, Research Presentation: Oral and Poster Presentations, Publishing Research Papers: Journals, Conferences, Indexing

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. New Age International. https://books.google.com/books?isbn=9788122415223
- Creswell, J. W., & Creswell, J. D. (2017). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE. https://books.google.com/books?isbn=9781506386706
- Kumar, R. (2019). *Research Methodology: A Step-by-Step Guide for Begin-ners*. SAGE. https://books.google.com/books?isbn=9781526449832
- Saunders, M., Lewis, P., & Thornhill, A. (2019). *Research Methods for Business Students*. Pearson. https://books.google.com/books?isbn=9781292208787
- Publication Manual of the American Psychological Association. https://books.google.com/books?isbn=9781433832161

Course Title: Network Administration	**L**	**T**	**P**	**Cr.**
Course Code: MCA3502	0	0	4	2

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Configure, manage, and troubleshoot network devices, including routers, switches, and firewalls.

CO2. Implement network security measures to protect against vulnerabilities and threats.

CO3. Monitor and optimize network performance using various tools and techniques.

CO4. Establish and maintain user access policies and authentication mechanisms.

CO5. Design and manage scalable networks to meet organizational requirements.

Course Content

1. Installation and configuration of server–client virtual lab using VMware/VirtualBox.
2. Server administration: users, groups, file permissions, SSH, sudo access, and service management.
3. IP addressing, subnetting, static IP configuration, gateway configuration, DNS resolver setup, and routing table management.
4. Network troubleshooting using command-line diagnostic tools such as `ping`, `ip`, `traceroute`, `ss`, `netstat`, `dig`, `nslookup`, and system logs.
5. Packet capture and protocol analysis using `tcpdump` and Wireshark.
6. Basic router and switch configuration using Cisco Packet Tracer.
7. VLAN configuration, trunking, inter-VLAN routing, static routing, OSPF, NAT, and ACL configuration.
8. DHCP server configuration, client testing, and troubleshooting.
9. DNS server configuration including forward lookup zone, reverse lookup zone, caching, and troubleshooting.
10. Web server and proxy server configuration and troubleshooting.
11. File sharing using Samba/NFS and secure file transfer using SFTP/FTPS.
12. Firewall configuration using `firewalld`/`ufw`, port-based access control, and rule testing.
13. Network monitoring and performance testing using `iperf3`, `mtr`, logs, and basic SNMP tools.

14. Backup, restore, disaster recovery planning, and scenario-based network troubleshooting.
15. SDN topology implementation using Mininet and OpenDaylight controller.

Suggested Readings:

- Nemeth, E., Snyder, G., Hein, T. R., Whaley, B., & Sandifer, D. (2017). *UNIX and Linux System Administration Handbook*. Addison-Wesley. https://books.google.com/books?isbn=9780134277554
- Limoncelli, T. A., Hogan, C. J., & Chalup, S. R. (2016). *The Practice of System and Network Administration*. Addison-Wesley. https://books.google.com/books?isbn=9780321919168
- Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer Networks*. Pearson. https://books.google.com/books?isbn=9780132126953
- Cisco Systems Official Training Materials. https://www.cisco.com/
- Kurose, J. F., & Ross, K. W. (2017). *Computer Networking: A Top-Down Approach*. Pearson. https://books.google.com/books?isbn=9780134085043

Course Title: Dissertation (Phase 1)	**L**	**T**	**P**	**Cr.**
Course Code: MCA3503	0	0	0	12

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Apply knowledge of recent computing technologies, skills and current tools of computer science and engineering.

CO2. Design and conduct experiments, as well as to analyze and interpret data.

CO3. Understand the contemporary research issues in the different areas of computer science & engineering.

CO4. Explore research gaps, analyze and carry out research in the specialized/ emerging areas.

Content Details

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations with implementation tools with suitable platform.

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning. Semester IV

Suggested Readings:

- Zobel, J. (2014). *Writing for Computer Science*. Springer. https://books.google.com/books?isbn=9781447166399
- Alley, M. (2018). *The Craft of Scientiftc Writing*. Springer. https://books.google.com/books?isbn=9781489979735
- IEEE Editorial Style Manual and Guidelines. https://journals.ieeeauthorcenter ieee.org/
- Google Scholar Search Index. https://scholar.google.com/
- LaTeX Project Official Reference Documentation. https://www.latex-project.org/

SEMESTER-IV

Course Title: Java Programming	L	T	P	Cr.
Course Code: MCA4550	3	0	0	3

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Describe the concepts of data types, variables, and arrays in object-oriented programming.

CO2. Analyze interfaces, class hierarchies, and exceptions in programs.

CO3. Construct appropriate diagrams and textual descriptions to communicate the AWT and Applet for web applications.

CO4. Implement the concept of SQL package, multithreading, and JDBC in Java.

CO5. Solve specified problems by using the Java SDK environment to create, debug, and run simple Java programs.

UNIT I **11 Hours**

Introduction to Java: Introduction to java, Java History, Java Features; How Java Differs from C and C++; Comments in java, Java Program Structure, Implementing a Java Program, Java Virtual Machine, Command Line Arguments, Programming Style, Java and Internet, Java and World Wide Web, Web Browsers; Hardware and Software Requirements; Java Support Systems, Java Environment, Java Tokens; Java Statements.

Constants, Variables and Data Types: Introduction; Constants, Variables, Data Types, Introduction to Operators, Expressions, Operator Precedence. Decision Making,

Branching and Looping: Decision making and branching Statements, Looping Statements, labeled loops, Jumping Statements.

UNIT II **11 Hours**

Classes, Objects and Methods: Introduction, defining a Class, Data member, member function, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members, Nesting of Methods. Arrays, Strings, Vectors: Arrays, Zagged Arrays, And Strings, String functions: Vectors, Wrapper Classes.

Inheritance: Extending a Class, Overriding Methods, Final Variables and Methods, Final Classes, Finalizer Methods, Abstract Methods and Classes, Visibility Control.

UNIT III **11 Hours**

Interfaces: Introduction, Defining Interfaces, Extending Interfaces, Implementing Interfaces, Accessing Interface Variables, Implementing Multiple Inheritance using Interfaces.

Packages: Introduction; System Packages, Using System Packages, Naming Conventions, Creating Packages, accessing a Package, Using a Package, Adding a Class to a Package, Hiding Classes.

Managing Errors and Exceptions: Introduction; Types of Errors; Exceptions, Exception Handling using Try, Catch and Finally block: Throwing Our Own Exceptions, Using Exceptions for Debugging.

UNIT IV **12 Hours**

Multithreading: Creating Threads, Thread Life Cycle, Synchronization, Thread Communication, Thread Priorities.

Applet Programming: How Applets Differ from Applications, Applet Life Cycle; Creating an Executable Applet.

JAVA Database Connectivity (JDBC): Merging data from Multiple tablets, joining, Manipulation: Database with JDBC, Prepared Statements, Transaction processing. Advanced Java Concepts: String handling and StringBuffer

Multithreading: Thread class, Runnable interface

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Schildt, H. (2018). *Java: The Complete Reference*. McGraw-Hill. https://books.google.com/books?isbn=9781260440232
- Bloch, J. (2018). *Effective Java*. Addison-Wesley. https://books.google.com/books?isbn=9780134685991
- Eckel, B. (2006). *Thinking in Java*. Prentice Hall. https://books.google.com/books?isbn=9780131872486
- Deitel, P., & Deitel, H. (2017). *Java How to Program*. Pearson. https://books.google.com/books?isbn=9780134743356
- Oracle Java SE Documentation Portal. https://docs.oracle.com/en/java/

Course Title: Java Programming Lab	**L**	**T**	**P**	**Cr.**
Course Code: MCA4551	0	0	2	1

Total Hours: 30

After completion of this course, the learner will be able to:

CO1. Classify the structure and model of the Java programming language.

CO2. Implement the given problems in Java programming language.

CO3. Develop software in the Java programming language.

CO4. Evaluate user requirements for software functionality required to decide whether the Java Programming Language can meet user requirements.

CO5. Connect Java programs to database using JDBC.

Course Content

1. Write a java program to find the Fibonacci series using recursive and non-recursive functions.
2. Write a java program to multiply two given matrices.
3. Write a java program that checks whether a given string is palindrome or not.
4. Write a java program for Method overloading and Constructor overloading.
5. Write a java program to represent Inheritance.
6. Write a java program to display the employee details using Scanner class.
7. Write a java program to represent Abstract class with example.
8. Write a java program to implement interface using extends keyword.
9. Write a java program to create user defined package.
10. Write a java program for creating multiple catch blocks.
11. Write an applet program that displays a simple message.
12. Write a Java program to compute factorial value using Applet.
13. Write a java program for handling Mouse events and Key events.
14. Write a java program that connects to a database using JDBC.
15. Write a java program for String manipulation exercises.
16. Write a java program to implement multithreading using Thread and Runnable.
17. Write a java program for file handling: reading from and writing to files.
18. Write a java program for building simple GUI applications using AWT/Swing components (buttons, text fields).

Suggested Readings:

- Schildt, H. *Java: The Complete Reference*. McGraw-Hill. https://books.google.com/books?isbn=9781260440232
- Oracle Java Language Tutorials. https://docs.oracle.com/javase/tutorial/
- GeeksforGeeks Java Tutorials. https://www.geeksforgeeks.org/java/
- Eclipse IDE Official Portal. https://www.eclipse.org/ide/
- IntelliJ IDEA Reference Documentation. https://www.jetbrains.com/idea/documentation/

Course Title: Mobile Application Development	**L**	**T**	**P**	**Cr.**
Course Code: MCA4552	0	0	4	2

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Identify the various concepts of mobile programming that make it unique from programming for other platforms.

CO2. Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces.

CO3. Program mobile applications for the Android operating system that use basic and advanced phone features.

CO4. Deploy applications to the Android marketplace for distribution.

UNIT I **15 Hours**

Mobile Application Development - Mobile Applications and Device Platforms

- Alternatives for Building Mobile Apps -Comparing Native vs. Hybrid Applications -The Mobile Application Development Lifecycle-The Mobile Application Front-End-The Mobile Application Back-End Key Mobile Application Services-What is Android-Android version History-Obtaining the Required Tools- Launching Your First Android Application-Exploring the IDE-Debugging Your Application-Publishing Your Application

UNIT II **15 Hours**

Understanding Activities-Linking Activities Using Intents-Fragments- Displaying Notifications Understanding the Components of a Screen- Adapting to Display Orientation-Managing Changes to Screen Orientation- Utilizing the Action Bar-Creating the User Interface Programmatically Listening for UI No-tifications

UNIT III **15 Hours**

Using Basic Views-Using Picker Views -Using List Views to Display Long Lists-Understanding Specialized Fragments - Using Image Views to Display Pictures -Using Menus with Views Using Web View- Saving and Loading User Preferences-Persisting Data to Files-Creating and Using Databases.

UNIT IV **15 Hours**

Sharing Data In Android-Creating Your Own Content Providers -Using The Content Provider SMS Messaging -Sending Email-Displaying Maps- Getting Location Data- Monitoring A Location.

Consuming Web Services Using HTTP-Consuming JSON Services- Creating Your Own Services - Binding Activities To Services -Understanding Threading. Advanced Mobile Development Concepts: Multimedia Integration: Audio,

Video, Camera, Location-Based Services And Google Maps API, Notifications And Alarms, Publishing Apps On Google Play Store / App Store Basics, Intro-duction To Cross-Platform Frameworks (Flutter, React Native).

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Phillips, B., Stewart, C., & Marsicano, K. (2019). *Android Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch. https://books.google.com/books?isbn=9780135242018
- Meier, R., & Lake, I. (2018). *Professional Android*. Wrox. https://books.google.com/books?isbn=9781118949528
- Android Developers Developer Training Guides. https://developer.android.com/courses
- Kotlin Language Reference Manual. https://kotlinlang.org/docs/home.html
- Google Material Design Spec. https://m3.material.io/

Course Title: Dissertation (Phase 2)	**L**	**T**	**P**	**Cr.**
Course Code: MCA4553	0	0	0	12

Total Hours: 45

After completion of this course, the learner will be able to:

CO1. Apply knowledge of recent computing technologies, skills and current tools of computer science and engineering.

CO2. Design and conduct experiments, as well as to analyze and interpret data.

CO3. Understand the contemporary research issues in the different areas of computer science & engineering.

CO4. Explore research gaps, analyze and carry out research in the specialized/ emerging areas.

Content Details

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations with implementation tools with suitable platform.

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Zobel, J. *Writing for Computer Science*. Springer. https://books.google.com/books?isbn=9781447166399
- Alley, M. *The Craft of Scientiftc Writing*. Springer. https://books.google.com/books?isbn=9781489979735
- IEEE Transactions and Journals Editorial Guidelines. https://ieeeauthorcenter.ieee.org/
- ResearchGate Academic Network. https://www.researchgate.net/
- Overleaf Online LaTeX Editor Portal. https://www.overleaf.com/

Course Title: Theory of Computation	**L**	**T**	**P**	**Cr.**
Course Code: MCA4554	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Recognize and comprehend formal reasoning languages.

CO2. Use the basic concepts of formal languages of finite automata techniques.

CO3. Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.

CO4. Design different types of Push down Automata as Simple Parser.

CO5. Analyze Context Free languages, Expression and Grammars.

UNIT I **13 Hours**

Introduction: Basic Terminology: Alphabet, Formal Language and operations on formal languages, Examples of formal languages.

Finite automata: Concept of Basic Machines, Properties and Limitations of Fi-nite State Machines, Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Equivalence of DFA and NDFA, Non-Deterministic Fi-nite automata with Λ- Transitions.

UNIT II **14 Hours**

Regular expression: Regular Languages and Regular Expressions, Kleen's Theorem. Arden's Method.

Properties of Regular sets: The Pumping Lemma for Regular sets, Application of the Pumping Lemma, Closure Properties of Regular Sets, Myhill- Nerode Theorem and Minimization of Finite Automata, Minimization Algorithm.

Finite Automata with output: Moore and Mealy Machines. Equivalence of Moore and Mealy Machines.

UNIT III **17 Hours**

Context Free Grammars: Examples and Definitions, Derivation trees and ambiguity, An Unambiguous CFG for Algebraic Expressions. Regular Grammar, Simplified forms and Normal forms: Removal of useless symbols and UNIT production, Removal of Λ-moves, Chomsky Normal Form (CNF), Greibach Normal Form (GNF).

Pushdown Automata: Introduction and Definition of Push-Down Automaton, Applications of Push-Down Automata.

UNIT IV **16 Hours**

Turing Machines: Definitions and Examples, Deterministic and Non- Deterministic Turing Machines, Unsolvable Problems: A Non-recursive Language and an Unsolvable Problem, PCP Problem and MPCP Problem.

More General Languages and Grammars: Recursively Enumerable and Recursive Languages, Unrestricted grammars, Context-sensitive language and grammar. Relation between languages of classes, Chomsky hierarchies of grammars.

Recursive And Recursively Enumerable Languages

Undecidability: Halting Problem, PCP (Post Correspondence Problem)

Transactional Mode: Lecture Method, E-Team Teaching, Video-based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage. https://books.google.com/books?isbn=9781133187790
- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation*. Pearson. https://books.google.com/books?isbn=9780321455369
- Linz, P. (2016). *An Introduction to Formal Languages and Automata*. Jones & Bartlett. https://books.google.com/books?isbn=9781284077247
- Mishra, K. L. P., & Chandrasekaran, N. (2008). *Theory of Computer Science: Automata, Languages and Computation*. PHI. https://books.google.com/books?isbn=9788120329683
- Kozen, D. C. (1997). *Automata and Computability*. Springer. https://books.google.com/books?isbn=9780387949079

Course Title: Principles and Practices of Management	**L**	**T**	**P**	**Cr.**
Course Code: MCA4555	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Understand the fundamental principles and functions of management in organizational contexts.

CO2. Apply management practices to plan, organize, lead, and control effectively.

CO3. Analyze real-world business scenarios to develop problem-solving and decision-making skills.

CO4. Demonstrate the ability to manage resources and foster teamwork for achieving organizational goals.

UNIT I **15 Hours**

Forms of Business Organizations and Ownership: Sole Proprietorship, Partnership, Joint Stock Company, Public & Private undertakings, Government Companies. Management: Meaning & Definition of Management, Nature, Scope and its various functions. Evolution of management thoughts: classical and new classical systems, contingency approaches, Scientific management.

UNIT II **15 Hours**

Planning: nature, purpose and functions, types of plan, Management by Objective (MBO), steps in planning. Decision Making: Meaning, Steps in Decision Making, Techniques of Decision Making. Strategic planning – concepts, process, importance and limitations; Growth strategies- Internal and external.

UNIT III **15 Hours**

Organizing: Concept, formal and informal organizations, task force, bases of departmentation, different forms of organizational structures, avoiding organizational inflexibility. Teamwork – meaning, types and stages of team building.

Concept of staffing– Recruitment and Selection.

Motivation – concept, importance and theories.

UNIT IV **15 Hours**

Authority: definition, types, responsibility and accountability, delegation, decentralization v/s centralization, determinants of effective decentralization. Line and staff authority.

Control: function, process and types of control, nature, process, significance and span of control. Direct control v/s preventive control.Trends and chal-lenges of management in global scenario, emerging issues in management: Introduction to Total Quality Management (TQM), Just in Time (JIT).

Transactional Mode: Lecture Method, E-Team Teaching, Video based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Koontz, H., & Weihrich, H. (2015). *Essentials of Management: An International, Innovation, and Leadership Perspective*. McGraw-Hill. https://books.google.com/books?isbn=9789339222864
- Robbins, S. P., Coulter, M., & DeCenzo, D. A. (2017). *Fundamentals of Management*. Pearson. https://books.google.com/books?isbn=9780134237473
- Prasad, L. M. (2020). *Principles and Practice of Management*. Sultan Chand & Sons. https://books.google.com/books?isbn=9789351611752
- Stoner, J. A. F., & Gilbert, D. R. (1995). *Management*. Pearson. https://books.google.com/books?isbn=9780131087477
- Harvard Business Review Management Channel. https://hbr.org/

Course Title: Software quality and testing assurance	**L**	**T**	**P**	**Cr.**
Course Code: MCA4556	4	0	0	4

Total Hours: 60

After completion of this course, the learner will be able to:

CO1. Explain and Apply Knowledge of Key Concepts of Software Testing, Quality and Testing Tools.

CO2. Draw the DD Graph and Identify the Various Test Cases from Paths of Flow Graph of Software Testing Problem and Determine the Complexity of Software.

CO3. Design Test Cases and develop a test suite, Write Test Scripts, and set environmental variables for Carrying Out the Various Levels of Testing Manually and Automatically.

CO4. Manage Software Defects and Risks Within a Software Project.

CO5. Work Effectively in Profile of Software Tester, Quality Assurance and Control officer, Project Manager and Leaders

UNIT I **15 Hours**

Software Testing Process, Objectives, Testing Techniques, Software Testing Life Cycle, Concept of Testing, Types of Errors, Stubs and Drivers, Verifica-tion and Validation, Different Types of Verification & Validation Mechanisms, Concepts of Software Reviews, Code Inspection and Code Walkthrough, Testing of Component-Based Software System, Energy Efficient Testing, Mobile Application Testing.

UNIT II **15 Hours**

Software Testing Methods, Testing Fundamentals, Test Case Design, White-Box Testing and its Types, Black-Box Testing and its Types, Software Testing Strategies, Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing, Test Planning, Budgeting and Scheduling.

UNIT III **15 Hours**

Software Testing Metrics, Concept and Developing Testing Metrics, Different Types of Metrics, Complexity Metrics, Defect Management, Definition of De-fects, Defect Management Process, Defect Reporting, Metrics Related to De-fects, Using Defects for Process Improvement.

UNIT IV **15 Hours**

Software Quality, Factors Affecting Software Quality, Quality Models, Software Quality Estimation, Quality Metrics, Quality Assurance, SQA Activities, Software Reviews, Formal Technical Reviews, Quality Control and Quality

Management, and SQA Plan. Quality Improvement, Pareto Diagrams, Cause-Effect Diagrams, Scatter Diagrams, Run Charts, Total Quality Management, Statistical Quality Assurance, Software Reliability, the ISO 9001 Quality Standard, Six Sigma, Informal Reviews.

Quality Costs, Quality Cost Measurement, Utilizing Quality Costs for Decision-Making. Manual Vs Automatic Testing, Basics of Automated Testing, Drawbacks of Manual Testing, Advantages of Automated Testing, Factors for Automation Testing, Types of Automation Testing Tools, Introduction to QTP, QTPIDE, Basic Components in QTP, QTP Framework, Write Scripts, Introduction to WinRunner, and Rational Robot.

Transactional Mode: Lecture Method, E-Team Teaching, Video-based learning, Demonstration, Peer Discussion, Open talk, Cooperative Teaching, Flipped Teaching, Collaborative Learning.

Suggested Readings:

- Naik, S., & Tripathy, P. (2008). *Software Testing and Quality Assurance: Theory and Practice*. Wiley. https://books.google.com/books?isbn=9780471789116
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing*. Wiley. https://books.google.com/books?isbn=9781118031964
- Patton, R. (2005). *Software Testing*. Sams Publishing. https://books.google.com/books?isbn=9780672327988
- Burnstein, I. (2003). *Practical Software Testing*. Springer. https://books.google.com/books?isbn=9780387951317
- International Software Testing Qualifications Board. https://www.istqb.org/