Nutan Maharashtra Vidya Prasarak Mandal's (NMVPM's)

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING AND TECHNOLOGY (NMIET)

Under Administrative Support - Pimpri Chinchwad Education Trust (PCET)

(An Autonomous Institute Approved by AICTE and Affiliated to SPPU, Pune)

Curriculum Structure and Syllabus
Of
First Year Master of Computer Applications (MCA)

Effective from Academic Year 2025-26

Nutan Maharashtra Vidya Prasarak Mandal's

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Under Administrative Support - Pimpri Chinchwad Education Trust

ESTD : 1908

Approved by AICTE | Autonomous Institute | Affiliated to SPPU | Accredited by NAAC

"Samarth Vidya Sankul", Vishnupuri, Talegaon Dabhade, Taluka Maval, District Pune - 410507

Tel. No. 02114 – 231666, E-mail :nmiettalegaon@gmail.com Web : www.nmiet.edu.in

Department of Computer Application (BCA & MCA)

Course Approval Summary: Board of Studies (MCA)

Sr. No.	Approved By	Signature and Stamp of Authority
1	Chairman, Board of Studies, MCA	Priyanka
2	Secretary, Academic Council, NMIET,Pune	
3	Chairman Academic Council, NMIET, Pune	Director Nutan Maharashtra Institute of Engineering & Technology Talegaon Dabhade - 410507

PREAMBLE

Dear Students, teachers and all stakeholders The field of computing is rapidly expanding and changing, especially, since the last decade with continuous emergence of new disruptive technologies such as artificial intelligence, data science, cyber security, Internet of things, robotics and so on. 21st Century has witnessed rapid technological developments in every sector including the field of Computing. Moreover, it has created new job roles and massive job opportunities for budding graduates. Premium Institutes, public and private Universities, autonomous and affiliated colleges in India have always played a crucial role in producing human resources with required skill sets by capturing and monitoring these developments and offered various UG and PG programmes. **Nutan Maharashtra Institute of Engineering and Technology (NMIET),** established in 2008 under the Nutan Maharashtra Vidya Prasarak Mandal, has consistently pursued its **primary objective** of providing quality technical education to the developing Maval region around Talegaon, Pune 3. As an AICTE-approved institution, NMIET has built a **strong reputation** for academic excellence and industry integration, with a recorded highest placement package of 23 LPA and an average of 3.5 LPA, attracting over 300 companies for campus recruitment.

The **rapidly evolving field** of computing has witnessed continuous emergence of disruptive technologies including artificial intelligence, data science, cybersecurity, Internet of Things, and robotics. These developments have created **new job roles** and massive employment opportunities for computing graduates. In response to these industry transformations and in alignment with the National Education Policy (NEP) 2020, NMIET is proud to introduce the **Bachelor of Computer Applications (BCA)** and **Master of Computer Applications (MCA)** programs starting from Academic Year 2025-26.

Program Overview and NEP 2020 Alignment

2.1 Credit Structure and Duration

- The **BCA program** follows a **six-semester structure** with a total of **120 credits**, averaging 20 credits per semester
- The **MCA program** builds upon undergraduate computing education with its own **80-credit framework** spread across four semesters
- Both programs incorporate the **Multiple Entry/Exit system** as recommended by NEP 2020, providing students with flexible academic pathways

Programme Outcomes (POs):

Learners are expected to know and be able to

PO1	Foundation Knowledge	Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving
PO2	Problem analysis	Identify, review, formulate and analyze problems for primarily focusing on customer requirements using critical thinking frameworks.
PO3	Development of Solutions	Design, develop and investigate problems with as an innovative approach for solution incorporating ESG/DSG goals
PO4	Modern Tool Usage	Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.
PO5	Individual and Teamwork	Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.
PO6	Project Management and Finance	Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.
PO7	Ethics	Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware
PO8	Life-long learning	Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

Course Summary

Department of Master of Computer Applications (MCA)

SEMESTER-WISE COURSE DISTRIBUTION

Course Distribution : Semester Wise				
Sr. No.	**Type of Course**	**No. of Courses / Semester**		**Total**
		1	**2**	
1	Basic Science Course	1	1	2
2	Program Core Course	3	2	5
3	Program Elective Course	0	2	2
4	Open Elective	0	0	0
5	Vocational and Skill Enhancement Course	0	0	0
6	Ability Enhancement Course	1	0	1
7	Entrepreneurship/Economics/ Management Course	0	0	0
8	Experiential Learning Courses	3	3	6
	Total	**8**	**8**	**16**

SEMESTER-WISE CREDIT DISTRIBUTION

Credit Distribution : Semester Wise				
Sr. No.	**Type of Course**	**No. of Credits / Semester**		**Total**
		1	**2**	
1	Basic Science Course	3	3	6
2	Program Core Course	9	6	15
3	Program Elective Course	0	6	6
4	Open Elective	0	0	0
5	Vocational and Skill Enhancement Course	0	0	0
6	Ability Enhancement Course	3	0	3
7	Entrepreneurship/Economics/ Management Course	0	0	0
8	Experiential Learning Courses	7	7	14
	Total	**22**	**22**	**44**

Curriculum Structure

First Year

MCA

First Year MCA Semester – I

CURRICULUM STRUCTURE

First Year MCA (With effect from Academic Year 2025-2026)															
Semester-I															
Course Code	**Course Name**	**Credit Scheme**				**Teaching Scheme (Hours/Week)**			**Evaluation Scheme and Marks**						**Total**
									FA		**SA**	**TW**	**PR**	**OR**	
		L	**P**	**T/A**	**Total**	**L**	**P**	**T/A**	**UT**	**CA**					
MCA01PC01	Software Engineering and Project Management	3	0	1	4	3	0	1	25	25	50	25	0	0	125
MCA01PC02	Data Structures Algorithm	3	0	0	3	3	0	0	25	25	50	0	0	0	100
MCA01PC03	Advance Database management System	3	0	0	3	3	0	0	25	25	50	0	0	0	100
MCA01BS01	Mathematical Foundation for Computer Application-1	3	0	1	4	3	0	1	25	25	50	25	0	0	125
MCA01AE01	Research Methodologies and IPR	3	0	1	4	3	0	1	25	25	50	0	0	0	100
MCA01PC02L	Data Structures Algorithm Lab	0	1	0	1	0	2	0	0	0	0	10	30	10	50
MCA01PC03L	Advance Database management System Lab	0	1	0	1	0	2	0	0	0	0	10	30	10	50
MCA01PC04L	Python Programming Lab	0	2	0	2	0	4	0	0	0	0	30	50	20	100
Total		**15**	**4**	**3**	**22**	**15**	**08**	**03**	**125**	**125**	**250**	**100**	**110**	**40**	**750**

L-Lecture, P-Practical, T/A-Tutorial/Activity, FA–Formative Assessment, SA-Summative Assessment, TW-Term Work, OR-Oral, PR-Practical, CA – Course Activity

First Year MCA (With effect from Academic Year 2025-2026)																
Semester-II																
Course Code	**Course Name**	**Credit Scheme**				**Teaching Scheme (Hours/Week)**			**Evaluation Scheme and Marks**							**Total**
									FA		**SA**	**TW**	**PR**	**OR**		
		L	**P**	**T/A**	**Total**	**L**	**P**	**T/A**	**UT**	**CA**						
MCA02PC01	Data Communication & Computer Network	3	0	1	4	3	0	1	25	25	50	25	0	0	125	
MCA02EL01X	Elective - I	3	0	0	3	3	0	0	25	25	50	0	0	0	100	
MCA02EL02X	Elective - II	3	0	0	3	3	0	0	25	25	50	0	0	0	100	
MCA02BS02	Optimization Techniques	3	0	1	4	3	0	1	25	25	50	25	0	0	125	
MCA02PC02	Software Testing	3	0	1	4	3	0	0	25	25	50	0	0	0	100	
MCA02EL01XL	Elective – I Lab	0	1	0	1	0	2	0	0	0	0	10	30	10	50	
MCA02EL02XL	Elective – II Lab	0	1	0	1	0	2	0	0	0	0	10	30	10	50	
MCA02PC03L	Advance Java Programming Lab	0	2	0	2	0	4	0	0	0	0	30	50	20	100	
Total		**15**	**4**	**3**	**22**	**15**	**08**	**03**	**125**	**125**	**250**	**100**	**110**	**40**	**750**	

L-Lecture, P-Practical, T/A-Tutorial/Activity, FA–Formative Assessment, SA-Summative Assessment, TW-Term Work, OR-Oral, PR-Practical, CA – Course Activity

List of Courses – Programme Elective Courses (First Year MCA)

FYMCA (Sem II)			
Course Code	**Elective 1**	**Course Code**	**Elective 2**
MCA02EL01A	Data Warehouse and Data Mining	MCA02EL02A	Artificial Intelligence
MCA02EL01B	Advance Web Development	MCA02EL02B	Cloud Computing
MCA02EL01C	Data Analytics	MCA02EL02C	Cyber Security and Laws
MCA02EL01AL	Data Warehouse and Data Mining Lab	MCA02EL02AL	Artificial Intelligence Lab
MCA02EL01BL	Advance Web Development Lab	MCA02EL02BL	Cloud Computing Lab
MCA02EL01CL	Data Analytics Lab	MCA02EL02CL	Cyber Security and Laws Lab

Syllabus of Courses
Semester I
First Year MCA

Program:	MCA							Semester : I
Course:	Software Engineering and Project Management							Code : MCA01PC01
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
4	3	-	1	25	25	-	50	100

Prerequisite:

1. Problem-solving and Analytical Thinking
2. Understanding of Computer Science Fundamentals Database Management Systems concepts is essential.

Course Objectives

This course aims at enabling students:

1. To understand fundamental principles and concepts of software engineering.
2. 2. To learn requirement analysis and system design principles.
3. 3. To study the process of Software Project Management for effective project planning.
4. 4. To acquire knowledge of Agile Project Management Framework.
5. 5. To apply Agile tools for software development.

Course Outcomes

After learning the course, the students should be able to:

1. Select and apply suitable software process models and requirement engineering techniques for given problems.
2. Create software design solutions using UML diagrams and apply principles of good user interface design.
3. Illustrate Software Project Management models for effective plan, manage and enhance projects.
4. Implement Agile methodologies to enhance project adaptability and responsiveness to changing requirements.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Overview of Software Engineering** Overview of Software Engineering, SDLC models, Requirement Engineering, Types of Requirements: -Functional and Nonfunctional, Four Phases of Requirement Engineering , Software requirement Specification (SRS), Structure and contents of SRS, IEEE SRS Format Case studies : based on SRS	12
2	**System Analysis and Modeling** Unified Modeling Language (UML): Use case Diagram, Activity Diagram, Sequence diagram, State Transition Diagram, Class Diagram and Object, Deployment Diagram, Graphical User	12

	Interface: Design patterns. Elements of good design, User Interface Design ,Case study on all above diagrams	
3	**Fundamentals of Project Management** Overview of project Management, Project management life cycle-IEEE Life Cycle , Quality Metrics ,Risk Management Process ,Linear Software Project Cost Estimation , COCOMO-I (Problem Statement) , Function Point Analysis (Problem Statement) ,The SEI Capability Maturity Model CMM ,Software Configuration management Case studies/Numerical Problems based on Risk management , COCOMO-I and FPA	**10**
4	**Agile Project Management Framework** Introduction and Definition Agile, Agile Project Life Cycle , Agile Manifesto: History of Agile and Agile Principles ,Team and roles of an Agile Team: Scrum Master Product Owner, Development Team , Key Agile Concepts: User stories, Story points , Techniques for estimating Story Points ,Product Backlog , Sprint Backlog, Product Vision and Product Roadmap , Sprint Velocity , Minimum Viable Product (MVP),Version and Release, Agile Project Management v/s Traditional Project Management , Agile Reports: Daily Reports, Sprint Burn down Chart and Reports User Stories Scenarios and writing user stories	**11**
	Total	**45**

Text Books:

1. Pressman, R. Software Engineering. McGraw-Hill, 2010.
2. Jacobson, I. Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1992.
3. Limaye, M.G. Software Testing Principles, Techniques and Tools. Tata McGraw-Hill, 2009

Reference Books

1. Jacobson, I. Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1992.
2. Bahrami, A. Object Oriented System Development. McGraw-Hill International Edition, 1999.
3. Rumbaugh, J., & Blaha, M. Object-Oriented Modeling and Design with UML. Pearson, 2004.
4. Larman, C. Agile and Iterative Development: A Manager's Guide. Addison-Wesley, 2003.
5. Beizer, B. Software Testing Techniques. DreamTech Press, 2nd Edition, 2003.
6. Patton, R. Software Testing. Sams Publishing, 2nd Edition, 2005.

E-Resources:

1. Agile:
   https://www.sealights.io/software-development-metrics/the-agile-process-scrum-kanban-and- x p/
2. UML Diagrams: Tutorials and Examples
3. https://www.geeksforgeeks.org/software-engineering-software-quality-assurance/
4. https://www.tutorialspoint.com/software_engineering/software_engineering_tutorial.pdf

Program:	MCA						Semester : I	
Course:	Data Structures Algorithm						Code : MCA01PC02	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Basic programming skills in C++
2. Understanding of mathematical logic and discrete structures
3. Familiarity with problem-solving methodologies

Course Objectives

This course aims at enabling students:

1. To understand the core principles and applications of fundamental data structures and algorithms.
2. To select and implement appropriate data structures for diverse computational problems.
3. To model and solve problems using linear and non-linear data structures such as arrays, linked lists, stacks, queues, hash tables, trees, heaps, and graphs.
4. To evaluate and compare searching, sorting, hashing, and heap algorithms for efficiency.

Course Outcomes

After learning the course, the students should be able to:

1. Define and explain the role of data structures and analyze algorithms using time and space complexity.
2. Apply and compare searching and sorting techniques to solve computational problems efficiently.
3. Construct and utilize linear data structures such as stacks, queues, and linked lists in problem- solving.
4. Implement and evaluate non-linear data structures like trees and graphs using appropriate traversal and storage techniques.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Foundations of Data Structures** Definition, role in problem-solving, types of data structures, review of arrays, algorithm analysis (time and space complexity, Big-O notation).	9
2	**Searching and Sorting** Searching methods: Linear Search, Binary Search, Hashing techniques; Sorting methods: Bubble, Selection, Insertion, Quick, Merge, Heap sort – performance analysis and use cases	9
3	**Linear Data Structures** Stacks – concepts, operations, and applications; Queues – standard, circular, and priority queues; Linked lists – singly, doubly, and circular linked lists, operations and applications.	9

4	**Non-Linear Data Structures** Trees – concepts, traversal techniques, binary trees, binary search trees, AVL trees, heaps; Graphs – representations (adjacency list/matrix), traversal algorithms (BFS, DFS), applications.	**9**
	Total	**36**

Text Books:

1. Tremblay Jean-Paul, Sorenson Paul G., An Introduction to Data Structures with Applications, McGraw Hill Publication, 2007, Mark Allen Weiss,
2. Samanta D., Classic Data Structures, PHI Publication, 2009

Reference Books

1. Srivastava S.K., Data Structures through C in Depth, BPB Publication, 2004
2. Lipschutz S., Schaum's Outlines Data Structures with C++, Tata McGraw Hill, 2019

E-Resources:

1. NPTEL: Data Structures and Algorithms (https://nptel.ac.in)
2. Data Structures in C++ – Scaler Topics (https://www.scaler.com/topics/course/cpp-data- structures/)

Program:	MCA						Semester : I	
Course:	Advance Database Management System						Code : MCA01PC03	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Database Fundamentals: Data models, database architecture, and database systems.
2. Relational Database Concepts: Relational algebra, relational calculus, and SQL is essential.

Course Objectives

This course aims at enabling students:

1. To understand the fundamental concepts and applications of Database Management Systems.
2. Design and Implement Concurrency Control Mechanisms:
3. Understand Database Recovery & Security Techniques:
4. Understand Parallel and Distributed Database Fundamentals.

Course Outcomes

After learning the course, the students should be able to:

1. Demonstrating the concept of fundamentals of relational database systems include: data models, database & DDBS architectures, and ER features.
2. Understand the concepts of transaction concurrency control, Query Processing and Security aspects.
3. Apply SQL & NoSQL development tools on different types of Schemas.
4. Demonstrate database design and Computation techniques for parallel and distributed database Technology.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Database Design, SQL Query Processing, Transaction & Concurrency Control** Introduction to Database, Data Models and Architecture of DBMS (Views of data: Schemas and Instances, Data Independence). Data Modelling using ER Diagram: Representation of Entities, Attributes, Relationships and their Types, Cardinality, Generalization, Specialization, Aggregation. Relational Data Model: Relational Database Model, Referential Integrity Constraints. Introduction to SQL (DDL, DML, Aggregate Functions and Joins).Transaction and Concurrency Control Concept of Transaction, ACID properties, Transaction States, Concurrency control.	9
2	**Database Recovery and Security Techniques** Failure Classification, Storage Structure, Recovery and Atomicity, Log-Based Recovery, Check Points, Shadow Paging. Introduction to Database backup, factors of database backups, Types of	9

	backups. Database Security in DBMS, Security Threats. Discretionary access control based on grant & revoking Privilege. Mandatory access control and role- based access control for Multilevel security.	
3	**Parallel and Distributed Database** Parallel Database System: Parallel Database Architectures; Parallel query processing and optimization; Load balancing; database clusters. Introduction to Distributed DBMS & Architecture, Characteristics. Distributed Data Processing, Promises of DDBMSs, Problem Areas. Distributed data storage (Fragmentation, Replication & Transparency).	**9**
4	**NOSQL database for Business Applications** Introduction to NOSQL Database: Overview, History of NoSQL Databases, The Definition of the Four Types of NoSQL Databases. Processing of NOSQL Column-Oriented NoSQL Databases using MongoDB, NoSQL Key/Value databases using MongoDB. Introduction to MongoDB Database, JSON and JSON Structure, Graph NoSQL Databases using Neo4J, NoSQL database development tools and programming languages.	**9**
	Total	**36**

Text Books:

1. Raghurama Krishnan, Johannes Gehrke, Database Management Systems, 3rd edition, Tata McGraw Hill, New Delhi, India
2. Introduction to database systems C.J. Date, Pearson.
3. Principles of Database Management James Martin, PHI
4. Elmasri Navate, Fundamentals of Database Systems, Pearson Education, India.
5. Sadalage, P. & Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Wiley Publications,1st Edition ,2019.
6. Principles of Distributed Database Systems, M.T. Ozsu and P. Valduriez, Prentice-Hall, 1991..
7. Distributed Database Systems, D. Bell and J. Grimson, Addison-Wesley, 1992.

Reference Books

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan Seventh Edition
2. Peter Rob, Carlos Coronel (2009), Database Systems Design, Implementation and Management, 7th edition
3. Dan Sullivan, "NoSQL For Mere Mortals", 1st Edition, Pearson Education India, 2015.
4. Dan McCreary and Ann Kelly, "Making Sense of NoSQL: A guide for Managers and the Rest of us", 1st Edition, Manning Publication/Dreamtech Press, 2013. (ISBN-13: 9789351192022)

E-Resources:

1. https://www.geeksforgeeks.org/sql-concepts-and-queries/
2. https://www.udemy.com
3. https://www.w3schools.com/sql/
4. https://www.codecademy.com/article/sql-commands
5. https://www.w3schools.com/sql/sql_intro.asp
6. https://www.javatpoint.com/sql-tutorial
7. https://www.geeksforgeeks.org/introduction-to-nosql/
8. https://www.edx.org/learn/nosql
9. http://libguides.regis.edu/tutorials.
10. https://www.mongodb.com/resources/basics/databases/nosql-explained
11. https://www.oracle.com/in/database/nosql/what-is-nosql/
12. https://www.javatpoint.com/nosql-databases
13. https://www.mysql.com/products/cluster/nosql.html

Program:	MCA						Semester : I	
Course:	Mathematical Foundation for Computer Application-1						Code : MCA01BS01	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
4	3	-	1	25	25	-	50	100

Prerequisite:

1. Basic understanding of mathematics, including set theory, algebra, and number systems.

Course Objectives

This course aims at enabling students:

1. Introduce foundational mathematical concepts essential for computer applications.
2. Explore probability theory as a basis for decision-making under uncertainty.
3. Analyze discrete and continuous probability distributions for modeling real-world scenarios.

Course Outcomes

After learning the course, the students should be able to:

1. Solve problems involving permutations, combinations, and counting principles.
2. Apply the Addition Rule, Inclusion–Exclusion Principle, and derangement concepts to practical problems.
3. Use conditional probability, independence, and Bayes' theorem in problem-solving.
4. Interpret and analyze discrete and continuous probability distributions such as Binomial, Poisson, Uniform and Normal.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Counting Principle:** Addition and multiplication rules, permutations of distinct objects, circular permutations, permutations with repetition.	12
2	**Principle of Inclusion and Exclusion:** Inclusion–Exclusion and Combinatorial Theorems Inclusion–Exclusion Principle and applications, derangements and their uses, integer solution problems, multinomial theorem with applications.	10
3	**Probability and Random Variables:** Probability Basics-Experiments, events, sample spaces; axioms of probability. Events-Independent and dependent events, conditional probability and its applications. Bayes' Theorem-Statement, proof, and applications. Random Variables-Discrete and continuous random variables; probability mass function (PMF), probability density function (PDF), and cumulative distribution function (CDF). Mathematical Expectation-Expectation, mean, variance, and higher moments. Bivariate Distributions-Discrete and continuous cases; joint probability distributions, marginal and conditional distributions, covariance, and correlation.	11

4	**Probability Distributions:** Discrete Probability Distributions Binomial Distribution – definition, properties, mean, variance. Poisson Distribution – definition, properties, mean, variance. Problem-solving using Binomial and Poisson distributions. Continuous Probability Distributions: Uniform Distribution – definition, properties, mean, variance. Normal Distribution – definition, properties, applications. Numerical problems based on Uniform and Normal distributions.	**12**
	Total	**45**

Text Books:

1. D. P. Apte, Probability and Combinatorics, Excel Books India, 2007.
2. S. C. Gupta, V. K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand & Sons, 2020.

Reference Books

1. Williams, A. S. W. A. S. *Statistics for business and economics*. South-Western,A TRIZ. 2011.
2. Judge, G. G., Griffiths, W. E., Hill, R. C., Lütkepohl, H., & Lee, T. C. *The theory and practice of econometrics* . John Wiley & Sons.1991.

E-Resources:

1. Statistical Methods Calculators (https://www.atozmath.com)
2. Discrete Mathematics – TutorialsPoint (https://www.tutorialspoint.com/discrete_mathematics/index.htm)

Program:	MCA						Semester : I	
Course:	Research Methodologies and IPR						Code : MCA01AE01	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
4	3	-	1	25	25	-	50	100

Prerequisite:

1. Basic understanding of mathematics, including set theory and algebra.
2. Fundamental knowledge of probability and statistical concepts.

Course Objectives

This course aims at enabling students:

1. To introduce students to the fundamentals and various types of research along with ethical issues and plagiarism.
2. To familiarize students with research tools and techniques including sampling, data collection, and statistical analysis.
3. To develop students' ability to write comprehensive research reports and use ICT tools for effective documentation.
4. To provide an understanding of Intellectual Property Rights (IPR) and related laws essential for protecting innovations.

Course Outcomes

After learning the course, the students should be able to:

1. Understand and explain the fundamental concepts and types of research, including ethical considerations and plagiarism.
2. Apply appropriate sampling techniques and statistical tools to analyze research data effectively.
3. Develop skills to prepare structured research reports and use ICT tools for documentation and referencing.
4. Demonstrate knowledge of Intellectual Property Rights (IPR) laws and their role in innovation and technology transfer.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Fundamentals of Research Methodology:** Meaning, objectives, and types of research (basic, applied, qualitative, quantitative), Characteristics of good research and formulation of research problems, Research process and research design (exploratory, descriptive, experimental), Review of literature and formulation of hypothesis- Ethical considerations and plagiarism	12
2	**Research Tools and Techniques:** Sampling methods: probability and non-probability sampling, Data collection techniques: primary and secondary data sources, Data classification, tabulation, graphical representation, Statistical measures: central tendency, dispersion, correlation, regression basics, Hypothesis testing: parametric and non-parametric tests (Chi-square, t-tests)	12

3	**Research Report Writing and Presentation:** Structure and components of research reports, theses, dissertations, writing abstracts, keywords, introduction, methodology, results, discussion, conclusion, referencing styles and bibliography (APA, MLA, Chicago),Use of ICT tools in research: software for data analysis and referencing (Mendeley, EndNote), Ethical issues in publication and research communication	**11**
4	**Intellectual Property Rights (IPR) and Related Laws:** Introduction to IPR: Copyrights, Patents, Trademarks, Designs, Geographical Indications, Copyright law: subject matter, ownership, rights, infringement, Patent law: patentable inventions, application process, rights, obligations, Trademark law and protection of industrial designs, Role of IPR in innovation and technology transfer, International IPR agreements (TRIPS, WIPO) and enforcement, Cyber laws and data protection related to IPR	**10**
	Total	**45**

Text Books:

1. D. P. Apte, Probability and Combinatorics, Excel Books, India, 2007.
2. S. C. Gupta, V. K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand & Sons, 2020.
3. C. R. Kothari, Research Methodology: Methods and Techniques, New Age International.
4. N. S. Gopalakrishnan, Intellectual Property Rights, Eastern Law Book House.
5. B. L. Wadehra, Law Relating to Intellectual Property, Universal Law Publishing.

Reference Books

1. Williams, A. S. W., Statistics for Business and Economics, South-Western, 2011.
2. Judge, G. G., Griffiths, W. E., Hill, R. C., Lütkepohl, H., & Lee, T. C., The Theory and Practice of Econometrics, John Wiley & Sons, 1991.
3. Kothari, C. R., Research Methodology: Methods and Techniques, Wiley Eastern Ltd.
4. V. K. Ahuja, Intellectual Property Rights in India, LexisNexis Butterworths Wadhwa, 2019.

E-Resources:

1. Statistical Methods Calculators – https://www.atozmath.com
2. Discrete Mathematics Tutorials – https://www.tutorialspoint.com/discrete_mathematics/index.htm
3. Research Methodology Tutorials – https://www.managementstudyguide.com/research- methodology.htm
4. Intellectual Property Rights (WIPO) – https://www.wipo.int/about-ip/en/
5. Mendeley: https://www.mendeley.com
6. EndNote: https://endnote.com

Program:	MCA						Semester : I
Course:	Data Structure Algorithm Lab						Code : MCA01PC02L
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	0	2	-	10	10	30	50

Prerequisites:

1. Knowledge of programming fundamentals.
2. Understanding of basic mathematical concepts.

Course Objectives

1. To apply the concepts of data structures for solving computational problems.
2. To implement and analyze various data structures and algorithms.
3. To understand the time and space complexity of algorithms.
4. To enhance problem-solving and logical thinking skills through practical implementation.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Implement linear and non-linear data structures.
2. Apply different searching and sorting techniques.
3. Demonstrate the use of recursion in problem-solving.
4. Analyze algorithm performance in terms of time and space complexity.
5. Implement stack, queue, linked list, tree, and graph operations in real-world scenarios.

Guidelines:

Students will be assessed based on:
The practical work done by them throughout the semester.
The Term work ,oral and practical exam have the weightage of 10 marks and 10 marks and 30 marks respectively.

Eclipse IDE setup required.

Detailed Syllabus

Assign. No.	Suggested List of Assignments (any 4 from both parts)	
1	Write a program to perform insertion, deletion, searching and transversal in 1D array.	PART A
2	Implement stack using arrays with push, pop, peek and display operations.	
3	Write a program to convert an infix expression into postfix and evaluate it.	
4	Implement queue using arrays with enqueue, dequeue and display operations.	
5	Implement circular queue using arrays.	

6	Implement singly linked with the following operations-insert at beginning, end and at a given position.	
7	Implement doubly linked list with insertion, deletion and transversal operations in both directions.	**PART B**
8	Implement circular linked list with insertion, deletion operations.	
9	Write a program to create a Binary Search Tree (BST) to perform inorder, preorder and postorder transversal	
10	Implement BFS for graph traversal.	
11	Implement the following sorting algorithms-bubble sort, selection sort, insertion sort, merge sort and quick sort	
12	Write a program to find factorial of number using recursion	

References:

1. Sahni, S. Data Structures, Algorithms, and Applications in C++. Universities Press.
2. Weiss, M.A. Data Structures and Algorithm Analysis in C++. Pearson Education.
3. Horowitz, E., Sahni, S., Mehta, D. Fundamentals of Data Structures in C. Universities Press.
4. Goodrich, M., Tamassia, R. Data Structures and Algorithms in Java. Wiley.
5. GeeksforGeeks — https://www.geeksforgeeks.org/
6. Programiz — https://www.programiz.com/dsa

Program:	MCA						Semester : I
Course:	Advance Database Management System Lab						Code : MCA01PC03L
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	0	2	0	10	10	30	50

Prerequisites:

1. Basic knowledge of Database Management Systems.
2. Understanding of SQL and basic database concepts is essential.

Course Objectives

1. To apply advanced concepts of database systems in practical applications.
2. To design and implement efficient database solutions for complex data requirements.
3. To explore distributed, NoSQL, and advanced database technologies.
4. To perform query optimization and transaction management.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Design and implement complex database schemas.
2. Use advanced SQL features such as views, triggers, stored procedures, and functions.
3. Implement transaction control and concurrency mechanisms.
4. Integrate NoSQL databases with applications.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Students are advised to use:

- Notebook
- MySQL, Neo4J
- Database tools - MySQL Workbench, pgAdmin, MongoDB Compass

Detailed Syllabus

Assign. No.	Suggested List of Assignments
1	Implement Joins, Nested Queries, Set Operations, Views, Indexes, Stored Procedures, Triggers, Cursors.
2	Implementing Checkpoints in DBMS.
3	Implement Discretionary Access Control (DAC) to demonstrate granting and revoking privileges in a DBMS.
4	Implement horizontal and vertical fragmentation by partitioning tables:

	• Horizontal: Rows based on region. • Vertical: Columns for specific departments.
5	Implement a view that integrates multiple fragmented tables.
6	Implement various types of backups (Full Backup/ Incremental Backup/ Differential Backup).
7	Implement CRUD operations using MongoDB/Cassandra/Neo4j.
8	Implement graph queries using Neo4j.

References:

1. Ramakrishnan, R., Gehrke, J. Database Management Systems. McGraw-Hill.
2. Elmasri, R., Navathe, S.B. Fundamentals of Database Systems. Pearson.
3. Silberschatz, A., Korth, H.F., Sudarshan, S. Database System Concepts. McGraw-Hill.
4. Chodorow, K. MongoDB: The Definitive Guide. O'Reilly Media.
5. Robinson, I., Webber, J., Eifrem, E. Graph Databases. O'Reilly Media.
6. GeeksforGeeks — https://www.geeksforgeeks.org/
7. MongoDB Documentation — https://www.mongodb.com/docs/

Program:	MCA					Semester : I	
Course:	Python Programming Lab					Code: MCA01PC04L	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
2	0	4	-	40	20	40	100

Prerequisite:

1. Computer Fundamentals
2. Basics Concepts of Programming is essential.

Course Objectives

This course aims at enabling students:

1. To solve real-world problems by applying programming concepts.
2. To develop and use functions and modules in Python for better code organization and reusability.
3. Develop desktop and command-line applications with Python for various purposes.
4. Present and demonstrate proficiency in Python programming through projects that apply concepts learned in the course

Course Outcomes

After learning the course, the students should be able to:

1. To learn and apply basic constructs of python such as data, operations, conditions, loops, data types.
2. To understand the use of functions, modules and concepts of exception handling apply it for solving the complex problems.
3. To develop Python programs that incorporate OOPS concept, regular expressions and multithreading for complex problem-solving and performance enhancement.
4. To implement various types of database operations in MongoDB

Guidelines:

Students will be assessed based on:
The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 80 marks and oral exam has the weightage of 20 marks.
Students are advised to use:
Latest Python IDLE

Detailed Syllabus		
Unit No.	**Description**	**Duration (Hrs.)**
1	**Fundamentals of Python** Introduction, Keywords, Identifiers, Literals, Operators, Data Types-Python blocks, Control flow, Loops, Loop manipulation using pass, continue, break and else, for loop using ranges, string, list and dictionaries, Programming using Python conditional and loops block, Comprehensions on List, Tuple, Dictionaries	9
2	**Functions, Modules & Packages, Exceptional Handling** Function Basics-Scope, nested function, Built-in functions, Types of functions, Decorators and Generators, Modules, Python built-in modules, Packages in Python, Exception Handling. Safeguarding file operation using exception handling	9
3	**Python Object Oriented Programming** Concept of class, object and instances, method call, Constructor, class attributes and destructors, Inheritance, super class, method overriding, Overloading operators, Static and Class methods, Python Regular Expression, Pattern matching and searching using regex in python, Multithreading.	9
4	**Python database interaction using MongoDB** Introduction to NoSQL database, Types of NoSQL, SQL Vs NoSQL, Introduction to MongoDB with python, Installing MongoDB on Windows, Exploring Collections and Documents, Performing CRUD Operations, Commit, Rollback and Cursor operation, Handling errors.	9
	Total	36

Assign. No.	**Suggested List of Assignments (Any 10)**
1	Write a program to classify numbers in a list as prime, even, or odd.
2	Write a program to count word occurrences in a sentence and save results to a file.
3	Write a program to manage student records with add, update, delete, display, and search functionalities.
4	Write a recursive function to flatten a nested list.
5	Write a program to sort a list of dictionaries by age and name using lambda and sorted().
6	Write a program to perform arithmetic operations with user input and exception handling.
7	Write a program to calculate average scores from a CSV file and save results to a file.
8	Write a class Bank Account with deposit, withdraw, and balance-checking methods.
9	Write a Vehicle base class and Car and Motorcycle subclasses demonstrating method overriding.
10	Write a program to find students enrolled in a specific course using list comprehension.

11	Write a program to extract email addresses from a text using regular expressions.
12	Write a program to connect to MongoDB and perform CRUD operations on student data with error handling.

References:

1. Lutz, M. Learning Python. O'Reilly Media, (2013)
2. Dawson, M. Programming with Python: A User's Book. Cengage Learning, (2023)
3. Beazley, D. Python Essential Reference. Addison-Wesley Professional, (2009)
4. Python For Beginners : https://www.python.org/about/gettingstarted/
5. Python Tutorial **:** https://www.w3schools.com/python/default.asp

Syllabus of Courses
Semester II
First Year MCA

Program:	MCA						Semester : II	
Course:	Data Communication and Computer Network.						Code : MCA02PC01	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
4	3	-	1	25	25	-	50	100

Prerequisite:

1. Computer fundamentals and programming.
2. Operating systems and software applications

Course Objectives

This course aims at enabling students:

1. Understand Network Topologies
2. Understand Data Communication Fundamentals.
3. Troubleshoot Transmission Media and Switching Issues.
4. Understand Error Detection and Correction Techniques.

Course Outcomes

After learning the course, the students should be able to:

1. Students will be able to design and implement network topologies, including LAN, WAN, MAN, and wireless networks
2. Students will be able to apply the principles of data communication to analyze and design communication systems.
3. Students will be able to troubleshoot common issues related to transmission media and switching, including signal degradation and network congestion.
4. Students will be able to design and implement wireless communication systems, including cellular networks, Wi-Fi, and Bluetooth.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Network Topologies and Network Devices** Network Topologies: Introduction, Definition, Selection, Criteria, Types of Topology- i) Bus ii) Ring iii) Star iv) Mesh v) Tree vi) Hybrid Network Connecting Devices: Hub, Switch, Router, Repeater, Bridge, Gateway, Modem, Wireless Communication: Infrastructure, Satellite Communication, Infrared Communication, Broadcast Radio Wi-Fi, Microwave Communication, Bluetooth Technology. Domain Network Services (DNS) Domain Names, Authoritative Hosts, Delegating Authority, Resource Records, SOA records, DNS protocol, DHCP & Scope Resolution	12
2	**Fundamentals of Data Communication and Networking** Process of data communication and its components: Transmitter, Receiver, Medium, Message, Protocol. Protocols, Standards, Standard organizations, Bandwidth, Data Transmission Rate, Baud Rate and Bits per second. Modes of Communication (Simplex, Half duplex, Full Duplex).Analog Signal and Digital Signal, Analog and Digital Transmission: Analog to Digital, Digital to Analog Conversion Network Architecture: Peer To Peer, Client Server Network	11

3	**Transmission Media and Switching** Communication Media: Guided Transmission Media: Twisted-Pair Cable, Coaxial Cable Fiber - Optic Cable Unguided Media: Radio Waves, Microwaves, Infrared, Satellite Line-of-Sight Transmission Point to Point, Broadcast Multiplexing: Frequency Division Multiplexing, Time Division Multiplexing Switching: Circuit Switched Network, Packet Switched Network	**10**
4	**Network Security** Threats, Packet-filtering firewalls, Fire wall policies and rules, Common Problem with Packet Filtering, SSL – Secure Socket Layer, IPSec (Internet Protocol Security), Virtual Private Networks, Symmetric Key Signatures, Public key Signatures, Error Detection and Correction Types of Errors: Single Bit Error and Burst Error, Redundancy Error Detection: Longitudinal Redundancy Check (LRC), Vertical Redundancy Check (VRC), Cyclic Redundancy Check(CRC), Forward Error Correction: Forward error Correction IEEE standards: 802.1, 802.2, 802.3, 802.4, 802.5 Wireless LANs: 802.1 1 Architecture, MAC Sublayer, Addressing Mechanism ,Bluetooth Architecture: Piconet, Scatternet, Mobile Generations:IG, 2G, 3G, 4G and 5G	**12**
	Total	**45**

Text Books:

1. F. Behrouz, Data communications and Networking, Tata McGraw Hill, New Delhi, 2006
2. T. Andrew S., Computer Networks, PHI Learning Pvt Ltd, Delhi, 2013
3. Godbole, Data Communication and Networks, Tata McGraw Hill, New Delhi 2006

Reference Books

1. E. Douglas, Internetworking with TCP/IP Principles, Protocols and Architectures, PHI
2. Data Communication and Networking by D P Nagpal
3. Data Communication and Networking by Behrouz A Forouzan
4. Computer Communication and Networking by Narayana Vikrama

E-Resources:

1. Data Communication & Computer Network: https://www.tutorialspoint.com/data_communication_computer_network/index.htm

Program:	MCA						Semester : I	
Course:	Data Warehouse and Data Mining						Code:MCA02EL01A	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. DBMS
2. Data Structure

Course Objectives

This course aims at enabling students:

1. To understand fundamental concepts, techniques and design principles of data warehousing and data mining.
2. To enable students to understand, implement and evaluate various algorithms in data mining.
3. Understanding Data Warehousing Concepts: Studying data warehouse architectures, OLAP, and project planning aspects in building a data warehouse.
4. Data Mining Techniques: Introducing concepts, techniques, design, and applications of data mining, including classical algorithms.

Course Outcomes

After learning the course, the students should be able to:

1. Understand the fundamental concepts of data warehousing and data mining.
2. Apply data mining techniques to solve real-world problems.
3. Analyze and interpret data to identify patterns and relationships.
4. Design and implement data warehouses and data mining solutions.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Introduction to Data Mining** Data Mining: Definition, functionalities, classification of data mining systems, data mining task primitives, integration with databases or data warehouses, and issues in data mining Data Preprocessing: Data cleaning, integration, transformation, reduction, and discretization	9
2	**Association Rule Mining and Classification** Association Rule Mining: Market basket analysis, Apriori algorithm, and improved Apriori algorithm Classification: Decision trees, Bayesian classification, rule-based algorithms, and prediction methods	9
3	**Clustering and Data Mining Applications** Clustering: Types of data, categorization of major clustering methods, partitioning methods, hierarchical methods, and density-based methods Data Mining Applications: Business intelligence, data visualization, and recent trends in data mining	9

4	**Clustering and Classification** Introductions to Clustering and Classification, Input and Output Attributes, Naïve Bayes Classification, k-Nearest-Neighbor Classifiers (Lazy Learners), Clustering: Major Clustering Algorithms-Partition Clustering: k-means clustering, Issues with the k-means algorithm, Hierarchical clustering: Agglomerative clustering and Divisive clustering. A case study on finding efficient Clusters/classification on sample data set.	**9**
	Total	**36**

Text Books:

1. J. Han, M. Kamber, Data Mining: Concepts and Techniques.Morgan Kaufmann. (2011)
2. Ponniah, P. Data Warehousing Fundamentals. John Wiley & Sons. (2011)
3. Gupta G.K., Introduction to Data Mining with Case Studies.PHI. (2014)
4. Parteek Bhatia. Data Mining and Data Warehousing-Principles and Practical Techniques. Cambridge University Press. (2019)
5. Khurana B. S. Data warehouse and Data Mining, Vision publication (2021)

Reference Books

1. Kimball, R., Ross, M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. Wiley.(2013)
2. Richard J. Roiger: Data Mining: A Tutorial-Based Primer.CRC Press. (2017)

E-Resources:

1. Data Warehousing Tutorial : https://www.tutorialspoint.com/data_warehousing/index.htm
2. Data Mining Tutorial**:** https://www.geeksforgeeks.org/data-mining/

Program:	MCA						Semester : II	
Course:	Advanced Web Development						Code :MCA02EL01B	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Basic knowledge of HTML, CSS, and JavaScript.
2. Understanding of client–server architecture, HTTP, and JSON.
3. Familiarity with programming concepts and basic database operations.

Course Objectives

This course aims at enabling students:

1. To introduce students to the fundamentals of Node.js and enable them to build efficient, event- driven server-side applications.
2. To equip students with skills to design and implement RESTful APIs using Express.js framework for scalable web services.
3. To familiarize students with MongoDB and teach them how to perform CRUD operations and complex queries on NoSQL databases.
4. To develop students' ability to create dynamic, responsive single-page applications using AngularJS with effective form validation and routing.

Course Outcomes

After learning the course, the students should be able to:

1. Understand the core concepts and architecture of Node.js and its event-driven model for building scalable server-side applications.
2. Apply Express.js framework to develop RESTful APIs and manage middleware, routing, and session handling in web applications.
3. Demonstrate the ability to perform CRUD operations and data aggregation using MongoDB as a NoSQL database.
4. Create dynamic, single-page web applications using AngularJS with form validation, routing, and AJAX integration.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**NodeJS** Introduction to NodeJS, Advantages of NodeJS, Traditional Web Server Model, NodeJS Process Model, Modules in NodeJS, Creating Custom Modules, HTTP Module, Creating Web Server, Read the Query String, File System Module, URL Module, Buffer, Streams Node Package Manager (NPM), Installing packages locally and globally, Events in NodeJS, Events Module, Event Loop, File Upload, Nodemailer Module, Serving Static Files, Template Engine, Express.JS or Express.	9
2	**MongoDB** NoSQL Database, Advantages of NoSQL, Types of NoSQL Databases, what is MongoDB, Need of MongoDB, Advantages of MongoDB,	9

	RDBMS vs NoSQL, Installing MongoDB, Creating of Database, Collection, Document, Dropping of Database, Collection and Document, Inserting and Updating of Document, Querying of Document, Projection, Limiting Records, Sorting Records, Indexing, Aggregation	
3	**Server-Side** Development with Express (E): Introduction to the Express Framework. Installing and Testing Express. Creating a Node.js Express App. Restructuring an Express App. Creating Templates. Using Express Middleware Functions. Creating the List Page. Creating the Details Page. Creating the Edit Page. Creating the Add Page. Deleting Data.	**9**
4	**AngularJS** Forms, Validation and Routing Using Simple form, working with different input elements, Validation of inputs, Form Events CSS Classes, Creating Custom Validation, Introduction to Single Page Application (SPA), HTML template creation, use of routing to make SPA, AJAX using AngularJS, AngularJS Animation	**9**
	Total	**36**

Text Books:

1. Mario Casciaro, Luciano Mammino – Node.js Design Patterns, Packt Publishing, 3rd Edition, 2020
2. Andrew Mead – Learning Node.js Development, Packt Publishing, 1st Edition, 2018
3. Shannon Bradshaw, Eoin Brazil, Kristina Chodorow – MongoDB: The Definitive Guide, O'Reilly Media, 3rd Edition, 2019
4. Peter Membrey, Eelco Plugge, David Hows – MongoDB Basics, Apress, 1st Edition, 2014

Reference Books

1. Shyam Seshadri, Brad Green – AngularJS: Up and Running – O'Reilly Media – 2014
2. Brad Dayley – Learning AngularJS – Addison-Wesley Professional – 2014
3. Rodrigo Branas – AngularJS Essentials – Packt Publishing – 2014

E-Resources:

1. Node.js Official Documentation – https://nodejs.org/en/docs/
2. Node.js Tutorials by W3Schools – https://www.w3schools.com/nodejs/
3. MongoDB Official Documentation – https://www.mongodb.com/docs/
4. MongoDB University Free Courses – https://learn.mongodb.com/
5. W3Schools MongoDB Tutorial – https://www.w3schools.com/mongodb/
6. TutorialsPoint Express.js – https://www.tutorialspoint.com/expressjs/index.htm

Program:	MCA						Semester: II	
Course:	Data Analytics						Code: MCA02EL01C	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Basics of Database Management Systems
2. Knowledge of probability and statistics

Course Objectives

This course aims at enabling students:

1. To introduce the fundamental concepts, processes, and life cycle of data analytics.
2. To develop the ability to apply predictive and descriptive analytics techniques for real- world problem solving.
3. To enable learners to implement object segmentation, time series analysis, and feature extraction methods.
4. To provide hands-on experience in using R programming for data analysis, visualization, and statistical evaluation.

Course Outcomes

After learning the course, the students should be able to:

1. Explain the concepts, process models, and life cycle of data analytics.
2. Apply supervised, unsupervised, and association rule mining techniques to solve analytical problems.
3. Analyze datasets using object segmentation, decision tree methods, and time series forecasting.
4. Develop R programming solutions for data visualization, preprocessing, and statistical evaluation.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Introduction to Data Analytics** Analytics, Analytics Process Model, Analytical Model Requirements. Data Analytics Life Cycle overview. Basics of data collection, sampling, preprocessing and dimensionality reduction	9
2	**Predictive and Descriptive Analytics** Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms - K- Means. Association Rule Mining - Apriori algorithm	9
3	**Object Segmentation**: Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc. Time Series Methods: Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy and Analyze for prediction	9

4	**R Programming for Data Analysis:** Introduction to R - R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis - Visualization Before Analysis, Dirty Data, Visualizing a Single Variable, Examining Multiple Variables, Data Exploration Versus Presentation, Statistical Methods for Evaluation	**9**
	Total	**36**

Text Books:

1. Foster Provost, Tom Fawcett, Data Science for Business, O'Reilly Media, 2013.
2. Anil Maheshwari, Data Analytics, McGraw Hill Education, 2017.
3. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer, 2017.
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer, 2009.

Reference Books

1. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
2. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.
3. Hastie T., Tibshirani R., Friedman J., The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009.
4. Andy Field, Discovering Statistics Using R, Sage Publications, 2012.

E-Resources:

1. https://www.edx.org/learn/data-analytics
2. https://www.kaggle.com/

Program:	MCA						Semester : II
Course:	Data Warehouse and Data Mining Lab						Code : MCA02EL01AL
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	-	2	-	10	10	30	50

Prerequisites:

1. Basic knowledge of programming in Python or R, including data manipulation and analysis libraries.
2. Fundamental understanding of statistics, probability, and linear algebra concepts.

Course Objectives

1. To equip students with skills in data preprocessing, including cleaning, normalization, encoding, and dimensionality reduction techniques.
2. To enable students to understand and implement association rule mining using Apriori and its optimized variants for discovering meaningful patterns.
3. To develop the ability to construct, train, and evaluate classification models such as decision trees, naive Bayes, and k-nearest neighbors.
4. To introduce students to clustering techniques including k-means, hierarchical clustering, and density-based methods, and to interpret their results effectively.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Apply data preprocessing techniques including handling missing values, normalization, encoding, and dimensionality reduction to prepare datasets for analysis.
2. Implement association rule mining using Apriori algorithm and interpret the resulting patterns for market basket analysis.
3. Build and evaluate classification models such as Decision Trees, Naïve Bayes, and K-Nearest Neighbors to solve real-world prediction problems.
4. Perform clustering analysis using algorithms like k-Means, hierarchical clustering, and DBSCAN, and interpret cluster structures and validation metrics.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Students are advised to use:

Python (Anaconda with Jupyter), R (with RStudio), and Weka for comprehensive data mining and machine learning experiments.

Detailed Syllabus

Assign. No.	Suggested List of Assignments
1	Data Preprocessing Objective: Perform data cleaning, integration, transformation, and reduction. Tasks: Handle missing values using imputation techniques (mean, median, mode). Normalize and standardize the dataset.

	Encode categorical variables (label encoding / one-hot encoding). Apply dimensionality reduction (PCA). Dataset: UCI Iris / Titanic / any CSV with mixed data types.
2	Association Rule Mining using Apriori Objective: Discover association rules for market basket analysis. Tasks: Load transaction data in basket format. Use the Apriori algorithm to find frequent itemsets. Generate strong rules using support, confidence, and lift. Tools: Python (mlxtend), R (arules), or Weka. Dataset: Retail dataset (Groceries from UCI).
3	Improved Apriori Algorithm Objective: Implement and compare improved Apriori with original. Tasks: Implement Apriori with pruning optimizations. Compare performance with the standard algorithm. Visualize frequent itemsets and execution time. Tools: Python with custom logic or modified mlxtend.
4	Classification using Decision Trees Objective: Build a decision tree classifier and visualize it. Tasks: Train/test split of dataset. Train a Decision Tree model (e.g., ID3, C4.5). Evaluate accuracy, precision, recall. Visualize the tree using graphviz or matplotlib. Dataset: Iris, Titanic, or custom CSV.
5	Classification using Naïve Bayes Objective: Classify data using a Naïve Bayes classifier. Tasks: Train a Naïve Bayes model. Perform predictions and confusion matrix analysis. Compare results with decision tree classifier. Dataset: Spam/Ham dataset or Iris.
6	k-Nearest Neighbor Classification Objective: Implement and evaluate a KNN classifier. Tasks: Normalize features before KNN. Test with different values of k. Evaluate performance using cross-validation. Dataset: MNIST subset or Iris dataset.
7	Clustering using k-Means Algorithm Objective: Apply k-means clustering and interpret results. Tasks: Choose optimal k using Elbow method or Silhouette score. Visualize clusters using PCA or t-SNE for 2D projection. Analyze intra-cluster and inter-cluster distances. Dataset: Iris / synthetic data.
8	Hierarchical Clustering Objective: Perform hierarchical clustering (agglomerative and divisive). Tasks: Use dendrograms to choose number of clusters. Apply linkage methods (single, complete, average). Compare with k-means results. Tools: SciPy in Python or R's hclust.

9	Density-Based Clustering (DBSCAN) Objective: Identify clusters using DBSCAN. Tasks: Find core, border, and noise points. Compare with k-means and hierarchical methods. Tune ε (epsilon) and MinPts parameters. Dataset: Any 2D or geospatial dataset with noise.
10	Mini Project / Case Study – Real Data Application Objective: Perform end-to-end data mining on a real-world dataset. Tasks: Data preprocessing. Apply classification (Naïve Bayes / KNN). Perform clustering (k-means, hierarchical). Generate insights using visualization. Dataset ideas: Customer segmentation Student performance data E-commerce purchase behavior COVID-19 data clustering

References:

1. Han, J., Kamber, M., & Pei, J., Data Mining: Concepts and Techniques, Elsevier, 2011.
2. Aggarwal, C. C., Data Mining: The Textbook, Springer, 2015.
3. Tan, P.-N., Steinbach, M., & Kumar, V., Introduction to Data Mining, Addison Wesley, 2006.
4. Géron, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 2019.
5. Gupta, G. K., Intellectual Property Rights: Unleashing the Knowledge Economy, Eastern Economy Edition, 2008.
6. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J., Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016

Program:	MCA						Semester : II
Course:	Advanced Web Development Lab						Code : MCA02EL01BL
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	-	2	-	10	10	30	50

Prerequisites:

1. Basics Concepts of Programming
2. Basic knowledge of HTML, CSS, JavaScript

Course Objectives

1. To introduce students to the fundamentals of Node.js and enable them to build efficient, event- driven server-side applications.
2. To equip students with skills to design and implement RESTful APIs using Express.js framework for scalable web services.
3. To familiarize students with MongoDB and teach them how to perform CRUD operations and complex queries on NoSQL databases.
4. To develop students' ability to create dynamic, responsive single-page applications using AngularJS with effective form validation and routing.

Course Outcomes

After learning the course, the students should be able to:

1. Understand the core concepts and architecture of Node.js and its event-driven model for building scalable server-side applications.
2. Apply Express.js framework to develop RESTful APIs and manage middleware, routing, and session handling in web applications.
3. Demonstrate the ability to perform CRUD operations and data aggregation using MongoDB as a NoSQL database.
4. Create dynamic, single-page web applications using AngularJS with form validation, routing, and AJAX integration.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Detailed Syllabus

Assign. No.	Suggested List of Assignments (Any Ten)
1	Install Node.js and Express, run a basic server, and verify setup.
2	Design Order Form with a total price updated in real time, which contains name of five products and their prices. Create a bill amount for all the products and calculate GST on the billing amount and display total amount. Use AngularJS.
3	Write AngularJS filter to check whether the entered number in the textbox is prime or not.

4	Write AngularJS service which makes the square of the number when it is called.
5	Write AngularJS service which makes the addition/subtraction of two numbers entered in textboxes when Add/Sub button is clicked.
6	Write NodeJS code to implement web server to serve the different content based on URL
7	Write a program to implement MongoDB data models
8	Write a program to implement CRUD operations on MongoDB
9	Create an application for Customer / Students records using AngularJS
10	Create an add form to insert data into MongoDB.
11	Implement cookies and sessions in Express app
12	Develop a REST API with GET, POST, PUT, DELETE methods.
13	Display single record details from MongoDB on a details page.
14	Create a basic Express app with routes and responses.
15	Create a simple web application using AngularJS, NodeJS and MongoDB. For example, you may design Hotel Management System etc. with minimal functionalities.

References:

1. Shyam Seshadri, Brad Green – AngularJS: Up and Running – O'Reilly Media – 2014
2. Brad Dayley – Learning AngularJS – Addison-Wesley Professional – 2014
3. Rodrigo Branas – AngularJS Essentials – Packt Publishing – 2014

Program:	MCA					Semester : II
Course:	Data Analytics Lab					Code : MCA02EL01CL

Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	-	2	-	10	10	30	50

Prerequisites:

1. Basics of Database Management Systems
2. Knowledge of probability and statistics

Course Objectives

1. To introduce the fundamental concepts, processes, and life cycle of data analytics.
2. To develop the ability to apply predictive and descriptive analytics techniques for real- world problem solving.
3. To enable learners to implement object segmentation, time series analysis, and feature extraction methods.
4. To provide hands-on experience in using R programming for data analysis, visualization, and statistical evaluation.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Explain the concepts, process models, and life cycle of data analytics.
2. Apply supervised, unsupervised, and association rule mining techniques to solve analytical problems.
3. Analyze datasets using object segmentation, decision tree methods, and time series forecasting.
4. Develop R programming solutions for data visualization, preprocessing, and statistical evaluation

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Students are advised to use:

R software with RStudio IDE

Detailed Syllabus

Assign. No.	Suggested List of Assignments (Any Ten)
1	Import datasets into R from CSV, Excel, and database sources.
2	Perform data cleaning and preprocessing on a raw dataset.
3	Apply sampling techniques to create training and testing datasets.
4	Implement dimensionality reduction using Principal Component Analysis (PCA) in R.
5	Implement classification using Naive Bayes on a dataset.

6	Implement K-Nearest Neighbors (KNN) classification in R.
7	Perform linear regression and evaluate model performance.
8	Apply K-Means clustering to a dataset and visualize the clusters.
9	Perform hierarchical clustering using the agglomerative method.
10	Implement association rule mining using the Apriori algorithm.
11	Build a classification tree and regression tree using the rpart package
12	Apply pruning techniques to avoid overfitting in decision trees.
13	Perform time series forecasting using ARIMA in R.
14	Decompose time series data using STL and interpret seasonal, trend, and residual components.
15	Conduct exploratory data analysis (EDA) and visualize single and multiple variables in R.

References:

1. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
2. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.
3. Hastie T., Tibshirani R., Friedman J., The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009.
4. Andy Field, Discovering Statistics Using R, Sage Publications, 2012.

Program:	MCA						Semester : II	
Course:	Artificial Intelligence						Code : MCA02EL02A	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Basic knowledge of programming.
2. Understanding logic and probability.

Course Objectives

This course aims at enabling students:

1. To introduce the fundamentals, history, and applications of Artificial Intelligence.
2. To develop problem-solving skills using heuristic search techniques and problem-reduction methods.
3. To understand knowledge representation, learning approaches, and reasoning under uncertainty.
4. To explore the design, components, and applications of expert systems

Course Outcomes

After learning the course, the students should be able to:

1. Apply appropriate search algorithm for any AI problem.
2. Demonstrate knowledge representation mechanism.
3. Design software agent to solve a problem.
4. Solve complex problems and to provide decision-making ability using expert system.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Basic understanding of AI** AI vs Human Intelligence. History of AI- Timeline and major milestones, Turing Test, and early symbolic AI. Applications of AI- AI in healthcare, robotics, finance, education, defense, Intelligent Agents- Agent and environment, Rationality, PEAS description (Performance measure, Environment, Actuators, Sensors).	9
2	**Search Techniques** AI problem, search process, brute force search, depth-first search, breadth-first search, time and space complexities, heuristics search, hill climbing, best first search, A* algorithm and beam search. problem reduction AO* algorithm, constraint satisfaction, means end analysis	9
3	**Knowledge Acquisitions** Knowledge representation issues, knowledge representation mechanism Representing knowledge using rules, Horne clause, matching, control knowledge, Type of learning, Knowledge Acquisition, Early working machine learning, learning by induction	9

4	**Reasoning** Reasoning under uncertainty, types of reasoning's, abduction, inheritance, truth maintenance system Expert System: Introduction to expert system, building blocks of expert systems, knowledge engineering, Phases of expert system, characteristics of expert system and a case study	**9**
	Total	**36**

Text Books:

1. Stuart Russell, Peter Norvig – Artificial Intelligence: A Modern Approach – Pearson, 4th Edition, 2020.
2. David L. Poole, Alan K. Mackworth – Artificial Intelligence: Foundations of Computational Agents – Cambridge University Press, 2nd Edition, 2017.

Reference Books

1. Elaine Rich, Kevin Knight, Shivashankar B. – Artificial Intelligence – McGraw Hill Education,3rd Edition
2. Saroj Kaushik – Artificial Intelligence – Cengage Learning India, 1st Edition.

E-Resources:

1. AI Resources – IBM Developer: https://developer.ibm.com/technologies/artificial-intelligence
2. AI Guide – Microsoft Learn: https://learn.microsoft.com/en-us/ai/

Program:	MCA						Semester : II	
Course:	Cloud Computing						Code :MCA02EL02B	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Basic understanding of computer networks
2. Basic understanding of operating systems.

Course Objectives

This course aims at enabling students:

1. To learn the fundamental concepts of cloud computing.
2. To Explore impact and contribution of cloud technologies to various domains.
3. To understand the use of cloud based services and tools to develop and deploy applications
4. To Encourage innovative thinking to design simple cloud-based solutions for real-world problems.

Course Outcomes

After learning the course, the students should be able to:

1. State service models and cloud deployment models of Cloud computing.
2. Explain layered cloud architecture and virtualization.
3. Determine types of cloud storage and data security issues.
4. Develop applications and deploy them using Cloud service providers.(AWS,GCP)

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Cloud Services and Cloud Models** Introduction to Cloud, Cloud Computing vs. Cluster Computing vs. Grid Computing, Introduction to Cloud Service Models: IAAS, PAAS, SAAS Characteristics, Benefits and Applications, Comparison of SAAS, PASS and IAAS,XAAS- Anything as a Service – Storage as a service, Network as a Service, Database as a Service, Cloud Deployment Models-Public, Private, Hybrid, Cloud Platforms: Google Cloud Platform, Microsoft Azure, Salesforce, AWS.	9
2	**Cloud Computing Architecture and Storage** Cloud computing stack: Comparison with traditional computing architecture (client/server), Key components of cloud architecture: Frontend, Backend, and Middleware. Layered Cloud Architecture Design :NIST Cloud Computing Reference Architecture ,Advantages of Cloud Storage ,Cloud Storage Providers :S3.	9
3	**Understanding Abstraction and Virtualization** Basics of Virtualization & Hypervisors Concept, Types of Virtualizations: Server, Storage and Network, Advantages and Disadvantages of Virtualization Machine Image, Virtual Machine (VM)Open-Source Virtualization Technology Examples: VMware: Full Virtualization, Virtual Box	9

4	**Cloud Technologies, applications and security** Cloud Applications: Moving Applications to the Cloud, Microsoft Cloud Services, Google Cloud Applications, Amazon Cloud Services Cloud platforms integrate healthcare-specific tools: telemedicine apps and health record systems (AWS for Healthcare). Data privacy and security issues, Access control and authentication. Reputation, Risk, Authentication in Cloud Computing.	**9**
	Total	**36**

Text Books:

1. Thomas Erl, "Cloud Computing: Concepts, Technology & Architecture", Pearson, 2013.
2. Rajkumar Buyya, "Cloud Computing: Principles and Paradigms", Wiley, 2011.

Reference Books

1. Kailash Jauaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah, "Cloud Computing", Black Book, Dreamtech , 2014
2. Barrie Sosinsky, "Cloud Computing Bible", Wiley India Pvt. Ltd. 2012.
3. Prasant Kumar Pattnaik et.al., "Fundamentals of Cloud Computing",Vikas Publication House Pvt. Ltd.,first Edition 2015

E-Resources:

1. Cloud Computing Tutorial https://www.tutorialspoint.com/cloud_computing/cloud_computing_tutorial.pdf

Program:	MCA						Semester : II	
Course:	Cyber Security and Laws						Code : MCA02EL02C	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
3	3	-	-	25	25	-	50	100

Prerequisite:

1. Students should have basic computer literacy,
2. familiarity with operating systems and internet use.

Course Objectives

This course aims at enabling students:

1. Develop fundamental understanding of cyber security concepts and cyber law.
2. Educate students on security architectures, attack types, fraud detection, and secure operations.
3. Provide hands-on exposure to cyber investigation, digital forensics, and assessment of security needs.
4. Foster responsible digital citizenship and prepare learners for real-world cyber challenges.

Course Outcomes

After learning the course, the students should be able to:

1. Understand and explain cyber security principles and concepts, distinguishing between attacks, threats, and vulnerabilities.
2. Classify and analyze types of security architectures and cybercrimes.
3. Apply investigation and forensic techniques in cybercrime scenarios and explain pertinent legal aspects.
4. Implement safe digital practices, evaluate organizational security, and apply network monitoring solutions.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Introduction to Cyber Security** Fundamental concepts of cyber security Computer and internet basics protocols (FTP, Telnet), IP addressing Security features in Operating Systems (Windows, Linux) Security principles, cyber ethics	9
2	**Cyber Attacks & Defense Mechanisms** Types and classification of cybercrimes and cyber-attacks (stalking, phishing, spamming, software piracy, cyber terrorism, etc.) Security architectures, defense methods, vulnerability assessment Intrusion Detection/Prevention Systems Authentication (user/bio-metric), password and antivirus security	9
3	**Safe Digital Practices & Emerging Technologies** Web, Email, Cloud, IoT, Social Media security, Safe browsing, secure online transactions, social networking best practices	9

	Smartphone security: OS, updating apps, wallets, privacy settings, Virtual currencies, Block chain technology, security auditing	
4	**Cyber Law, Digital Forensics & Investigation** Cyber law basics: IT Act 2000, amendments, IPC, RBI Act, IPR Jurisdiction of cybercrime, evidentiary value of digital data (email, SMS) ,Creating awareness and healthy cyber practices, Digital forensics: branches, investigation methods, evidence management, criminal profiling	**9**
	Total	**36**

Text Books:

1. Cryptography, Network Security and Cyber Law by Bernard L. Menezes and Ravinder Kumar, Cengage Learning

Reference Books

1. Cyber Criminology: Exploring Internet Crimes and Criminal Behavior by K. Jaishankar, CRC Press
2. Computer Network Security and Cyber Ethics by Siva Ram Murthy, B.S. Manoj, McFarland & Company, INC
3. Data Communication and Networking by B. Forouzan, TMH
4. Cryptography and Network Security by William Stallings, Pearson Publication
5. An Official Guide to Ethical Hacking by Ankit Fadia, Trinity Publisher

E-Resources:

1. https://heimdalsecurity.com/pdf/cyber_security_for_beginners_ebook.pdf
2. http://larose.staff.ub.ac.id/files/2011/12/Cyber-Criminology-Exploring-Internet-Crimes-and- Criminal-Behavior.pdf

Program:	MCA						Semester : II
Course:	Artificial Intelligence Lab						Code : MCA02EL02AL
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	-	2	-	10	10	30	50

Prerequisites:

1. Basic knowledge of programming.
2. Understanding logic and probability.

Course Objectives

1. To introduce the fundamentals, history, and applications of Artificial Intelligence.
2. To develop problem-solving skills using heuristic search techniques and problem-reduction methods.
3. To understand knowledge representation, learning approaches, and reasoning under uncertainty.
4. To explore the design, components, and applications of expert systems.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Apply appropriate search algorithm for any AI problem.
2. Demonstrate knowledge representation mechanism.
3. Design software agent to solve a problem.
4. Solve complex problems and to provide decision-making ability using expert system.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Students are advised to use:

Latest Python IDLE

Detailed Syllabus

Assign. No.	Suggested List of Assignments (Any Ten)
1	Implement Depth First Search using Python.
2	Write a Program to demonstrate Breadth First Search using Python.
3	Write a program to Implement AO* algorithm
4	Write a Program to Implement Water-Jug problem using Python
5	Write a Program to demonstrate Best First Search using Python
6	Implement Tic-Tac-Toe game using Python.
7	Write a Program to Implement 8-Puzzle problem using Python

8	Implement N queen problem using Python.
9	Implement tower of Hanoi using Python.
10	Text Preprocessing and Tokenization for Natural Language Processing using Python Strings
11	Write a program in Python to simulate a simple chatbot for Turing Test using keyword matching
12	Develop a Python Decision Tree Classifier to Demonstrate Learning by Induction
13	Implement Rule-Based Knowledge Representation and Inference System in Python
14	Write a program in python to Implement a Basic Text Editor using Tkinter.
15	Write a program in python to create To-Do List Application using Tkinter.

References:

1. Elaine Rich, Kevin Knight, Shivashankar B. – Artificial Intelligence – McGraw Hill Education,3rd Edition
2. Saroj Kaushik – Artificial Intelligence – Cengage Learning India, 1st Edition.
3. Mark Lutz, Learning Python, O'Reilly Media, 5th Edition, 2013

Program:	MCA						Semester : II
Course:	Cloud Computing Lab						Code : MCA02EL02BL
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	-	2	-	10	10	30	50

Prerequisites:

1. Basic understanding of cloud computing concepts
2. Basic understanding of APIs, virtualization, and system administration.

Course Objectives

1. To utilize cloud service provider APIs and SDKs for cloud operations.
2. To understand and apply cloud migration strategies.
3. To manage and monitor resources in cloud environments.
4. To automate infrastructure provisioning and scaling.
5. To evaluate performance, cost, and reliability factors of cloud deployments.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Apply cloud services using API's/SDK's of providers like AWS, Azure, and GCP.
2. Understand and implement cloud migration strategies for transitioning applications, databases, and workloads from on premise to cloud environments using different tools.
3. Develop and Implement strategies for managing and monitoring cloud resources.
4. Applyautomationtechniquesforinfrastructureprovisioningandscalingusing cloud-native and third-party tools.
5. Assess and compare cloud deployments by analyzing performance, cost efficiency,reliability,andscalabilitytooptimizeoperationaleffectivenessand decision-making.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Students are advised to use:

1.AWS and GCP

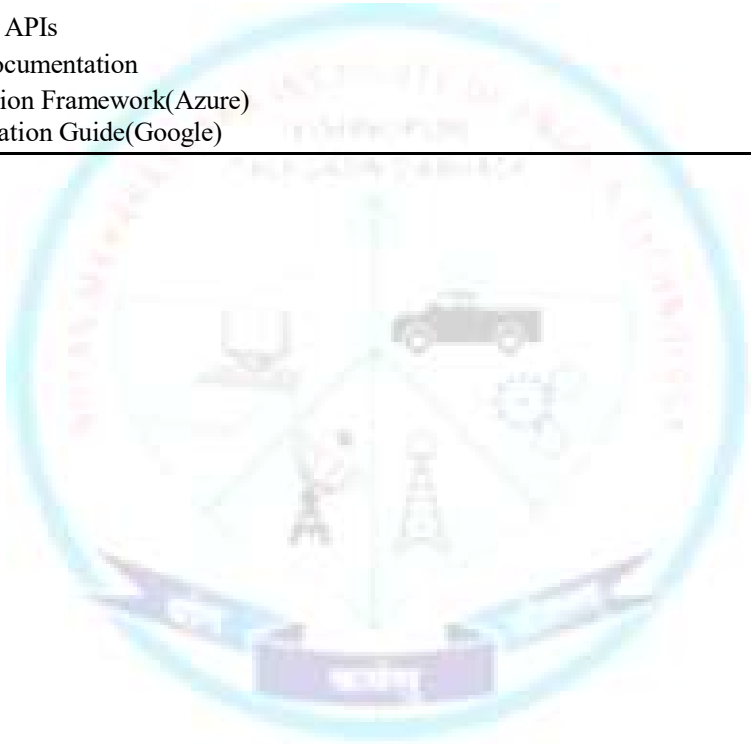
Detailed Syllabus

Assign. No.	Suggested List of Assignments (Any Ten)
1	Login with AWS account and explore AWS services.
2	Create and linux based EC2 instances by executing minimum 15 linux commands
3	Create an EC2 instance and configure it as web server and verify the output in the client system
4	Create and connect to instance using command line and using GUI based method
5	Deploy and test a program using AWS Lambda function

6	Monitor the analysis of lambda function using cloud watch monitoring tool
7	Create a S3 bucket to upload and download file
8	Building and Testing SOAP & REST APIs in AWS Cloud
9	Image Classification using Amazon Sage Maker (No-code via Studio Lab)
10	Sentiment Analysis using AWS Comprehend
11	Predict Future Values using Amazon Forecast
12	Object & Face Detection using AWS Recognition

References:

1. AWS Documentation
2. Google Cloud APIs
3. Azure REST APIs
4. Terraform Documentation
5. Cloud Adoption Framework(Azure)
6. Cloud Migration Guide(Google)



Program:	MCA						Semester : II
Course:	Cyber Security and Cyber Laws Lab						Code : MCA02EL02CL
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
1	-	2	-	10	10	30	50

Prerequisites:

1. Basic computer literacy: operating systems, browsing the internet, installing software.
2. Fundamentals of networking: LAN/WAN, IP addressing, common protocols.
3. Logical reasoning and ethical online practices

Course Objectives

This course aims at enabling students:

1. To provide students hands-on skills in cyber security concepts
2. network protection, cybercrime investigation, digital forensic
3. the essentials of cyber law for secure, responsible digital citizenship

Course Outcomes

After learning the course, the students should be able to:

1. Identify and analyze threats/vulnerabilities in digital systems.
2. Apply cybercrime investigation methods and forensic techniques.
3. Use security tools to safeguard networks, systems, and personal data.
4. Understand and relate cyber law concepts to practical cyber scenarios.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Detailed Syllabus

Assign. No.	Suggested List of Assignments
1	Operating System Security • Demonstrate configuring firewall settings and password policies in Windows or Linux.
2	Network Setup and Security • Set up a LAN or simulate network types (LAN, WAN), configure basic network security, and identify vulnerable points.

3	Internet Threat Analysis • Investigate, detect, and document at least two types of cyber-attacks (e.g., phishing, spam) using email or web simulations.
4	Authentication Methods Exploration • Implement different user authentication methods (password and biometric if available) and analyze their strengths and weaknesses.
5	Malware Investigation • Use an antivirus tool to scan a system, analyze findings, and suggest preventive measures. Prepare a short report on malware types encountered.
6	Safe Browsing Practices • Demonstrate safe online shopping or social media usage by highlighting security settings, privacy precautions, and reporting unsafe content.
7	Mobile Security Audit • Assess the security configuration of a smartphone. Install/update apps, enable device encryption, and test secure wallet/payment setup.
8	Incident Response Workflow • Perform a simulated incident response: detect an attack, isolate affected components, and document the steps for evidence preservation.
9	Digital Forensics Evidence Collection • Collect and submit digital evidence from a prepared scenario (e.g., log files, images) and prepare an evidence report suitable for legal proceedings.
10	Cyber Law Case Study Presentation • Select a real-world cybercrime case. Analyze the case with respect to the IT Act, digital evidence, and jurisdiction. Summarize findings in a presentation.

References:

1. Cryptography, Network Security and Cyber Law – Bernard L. Menezes & Ravinder Kumar
2. Cyber Criminology: Exploring Internet Crimes and Criminal Behavior – K. Jaishankar
3. Computer Network Security and Cyber Ethics – Siva Ram Murthy & B.S. Manoj
4. Data Communication and Networking – B. Forouzan
5. An Official Guide to Ethical Hacking – Ankit Fadia
6. Cryptography and Network Security – William Stallings

Program:	MCA						Semester : II	
Course:	Optimization Techniques						Code : MCA02BS02	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
4	3	-	1	25	25	-	50	100

Prerequisite:

1. Basic mathematical knowledge is essential.

Course Objectives

This course aims at enabling students:

1. To understand the role and principles of optimization techniques in business world.
2. To understand the process of problem statement formulation of the business scenario.
3. To understand the implementation of various decision-making techniques in the process of decision making.
4. To gain the techniques and skills on how to use optimization techniques to support the decision making in business world.

Course Outcomes

After learning the course, the students should be able to:

1. Understand and formulate linear programming models to solve optimization problems in various business contexts.
2. Apply sequential models to make informed decisions in dynamic and uncertain environments. Perform unit, integration, system, and acceptance testing.
3. Utilize Markov chains and simulation techniques to model.
4. Apply PERT/CPM techniques to plan, schedule, and control projects effectively, including managing replacement decisions.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Linear Programming & Transportation Problems** Definitions of Linear Programming, Basic Theorems and Properties, Advantages and Limitations, Applications of Linear Programming, Concept of Simplex Method with Problems (No Graphical Solutions), Transportation Problem – North-West Corner Method (NWCM), Least Cost Method (LCM), Vogel's Approximation Method (VAM), Optimization using MODI Method	12
2	**Sequencing & Queuing Models** Sequencing – Processing n jobs through 1 machine, 2 machines, 3 machines, Queuing Theory – Characteristics of Queuing Models, Transient and Steady States, Model I: (M/M/1) :(FCFS/∞/∞), Miscellaneous Problems on Sequencing and Queuing	10
3	**Markov Chains & Simulation Techniques** Markov Chains – Applications in technical/functional areas, Steady State Probabilities and Implications, Decision-making using Markov Models, Simulation Techniques – Monte Carlo Approach, Miscellaneous Problems	11

4	**Project Management & Decision/Game Theory** PERT vs CPM – Differences, Network Diagrams, Time Estimates (Forward Pass, Backward Pass), Critical Path, Probability of Meeting Deadlines, Floats and Slack, Calculations on CPM and PERT Networks, Decision Theory – Introduction, Steps of Decision-Making Process, Decision under Uncertainty and Risk, Game Theory – Introduction, n×m Zero Sum Games with Dominance, Solutions using Algebraic, Arithmetic and Matrix Strategy	**12**
	Total	**45**

Text Books:

1. Operations Research by Panneerselvam
2. Operations Research Theory and Application by J. K. Sharma –Mac-Millan Publication
3. Statistical and Quantative Methods – Mr. Ranjit Chitale

Reference Books

1. Statistical Methods – S.P.Gupta, Sultan Chand, New Delhi
2. Operation Research by V. k. Kapoor
3. Operations Research by Kanti Swaroop, P. K. Gupta and Man Mohan
4. Introduction to Operations Research by Hiller & Lieberman, Tata Mc GrawHill
5. Operations Research by H. A. Taha
6. Operation Research by Hira & Gupta

E-Resources:

1. www.orsi.in
2. www.atozoperationalresearch.com

Program:	MCA						Semester : II	
Course:	Software Testing						Code : MCA02PC02	
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	FA		TW	SA	Total
				UT	CA			
4	3	-	1	25	25	-	50	100

Prerequisite:

1. Basic programming in Java/Python/C++
2. Knowledge of Software Development Life Cycle (SDLC)
3. Familiarity with databases and web applications

Course Objectives

This course aims at enabling students:

1. To introduce fundamental concepts of software testing.
2. To apply different testing methodologies for software quality assurance.
3. To use automated testing tools for functional and performance testing.
4. To understand standards and practices for software quality improvement

Course Outcomes

After learning the course, the students should be able to:

1. Explain testing principles, techniques, and their applications.
2. Design test cases using black-box and white-box testing.
3. Perform unit, integration, system, and acceptance testing.
4. Apply automated tools like Selenium, JUnit, TestNG for testing.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Foundations of Software Quality** Quality, Quality Assurance (QA), Quality Control (QC), Difference between QA and QC, Software Quality Assurance Challenges, Software Quality Factors, Software Reliability, Reliability Metrics – ROCOF, MTTF, MTTR, MTBF, Availability, SQA Standards – ISO 9000	10
2	**Software Testing Fundamentals** Definition & Objectives of Testing, Role of Testing in Software Quality, Causes of Software Failure – Error, Bug, Fault, Defect, Failure, Economics of Testing, Seven Principles of Testing, Software Testing Life Cycle (STLC), Verification & Validation – V Model, W Model, Agile Testing & Test-Driven Development (TDD), Levels of Testing – Unit Testing, Integration Testing, System Testing, User Acceptance Testing (UAT), Functional Testing (Black-box), Structural Testing (White-box), Non- functional Testing, Regression Testing, Performance Testing, Usability Testing, Security Testing, Portability Testing, Smoke Testing, Sanity Testing	12
3	**Static Testing & Test Management** Static Techniques – Reviews, Walkthroughs, Inspections, Static Analysis Tools, Test Organization – Roles of Tester, Test Lead, Test Manager, Test Planning – IEEE 829 Standard Test Plan, Test Cases – Positive and Negative, Test Log, Test Summary Report, Defect Density, Defect Life Cycle, Incident/Defect Report, Risk-based Testing – Project Risk, Product Risk, Configuration Management for Testing	12

4	**Tool Support & Automation** Types of Test Tools – CAST, Benefits and Risks, Selenium WebDriver, TestNG, JMeter, Postman, ETL Testing Tools, JIRA for Project Management, Basics of Automation Frameworks – Data-driven Testing, Hybrid Testing	**11**
	Total	**45**

Text Books:

1. Foundations of Software Testing by Rex black, Erik Van Veenendaal, Dorothy Graham
2. (2020)-Cengage Learning: London UK, 5th Edition
3. Software Engineering by Sommerville-Pearson,8thEdition
4. Daniel Galin, "Software Quality Assurance: From Theory to Implementation", Pearson
5. Addison-Wesley, 2012. 2.
6. Effective Methods for Software Testing by William Perry- Wiley Pub, 3rd Edition.

Reference Books

1. Ron Patton, *Software Testing*, Pearson.
2. Aditya P. Mathur, *Foundations of Software Testing*, Pearson.
3. Roger S. Pressman, "Software Engineering-A Practitioner's Approach", McGraw Hill pub.2010
4. Software Testing Techniques by Boris Beizer-DreamTech Pub,2nd Edition

E-Resources:

1. https://www.ministryoftesting.com/articles/99-essential-resources-to-help-software-testers
2. https://github.com/PaulWaltersDev/FreeLearningResourcesForSoftwareTesters
3. https://huddle.eurostarsoftwaretesting.com/resources/
4. https://www.geeksforgeeks.org/software-testing/software-testing-tutorial/

Program:	MCA					Semester : II	
Course:	Advance Java Programming Lab and Software Testing Lab					Code : MCA02PC03L	
	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
Credits	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
2	0	4	-	40	20	40	100

Prerequisites:

1. Basic knowledge of Java programming, including object-oriented concepts.
2. Understanding of web technologies such as HTML, HTTP, and client-server architecture.

Course Objectives

1. To introduce the concepts of Java Generics and Collections for efficient data handling.
2. To develop skills in building and deploying Servlets for dynamic web applications.
3. To enable students to create interactive web pages using Java Server Pages (JSP) and connect to databases.
4. To provide understanding of Spring MVC framework and its components for designing scalable web applications.

Course Outcomes

On completion of the course, learners should be able to understand the

1. Understand and apply Java Generics and the Collection Framework for effective data management.
2. Develop and deploy Java Servlets to create dynamic web applications with session and cookie management.
3. Design web pages using JSP components and implement database connectivity through JDBC.
4. Build web applications using Spring MVC and Spring Boot frameworks, incorporating dependency injection and MVC architecture.

Guidelines:

Students will be assessed based on

The practical work done by them throughout the semester, the Practical Exam, and the Oral Exam. Practical work and practical exam collectively have the weightage of 80 marks and oral exam has the weightage of 20 marks.

Students are advised to use:

1. Java development environment with JDK, an IDE Eclipse or IntelliJ IDEA, Apache Tomcat server for Servlet and JSP deployment.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Collection and Generic :** Introduction to Generics , Generics Types and Parameterized Types, Wildcards , Java Collection Framework: Collections (Basic Operations, Iteration) List, Set, Maps, Queues, Arrays Lambda Expressions - Lambda Type Inference, Lambda Parameters, Lambda Function Body, Returning a Value, From a Lambda Expression, Lambdas as Objects.	9

2	**Servlets** Fundamentals of Java Servlet programming, A simple java Servlet , Servlet life cycle , Developing and Deploying Servlets, Working with cookies	**9**
3	**Java Server Pages** JSP Overview-Installation- JSP Tags-Components of a JSP page Expressions Script lets-Directives, JSP object, JDBC connectivity	**9**
4	**Spring MVC** Overview of the Spring Framework, Spring MVC Annotation, Spring MVC Architecture, Spring MVC Flow, Spring Form Handling, Spring Core and Spring Boot Dependency injection and inversion of control (IoC)	**9**
	Total	**36**
Asgn. No.	**Suggested List of Assignments**	
1.	Create a simple servlet that prints "Hello, World!" on a web browser. Use HttpServlet class and override doGet() method.	PART A
2.	Develop a servlet that logs messages for init(), service(), and destroy() methods. Observe the execution order by accessing the servlet multiple times.	
3.	Create a servlet to accept user details (name, email, age) through a form and display them on a response page.	
4.	Create a servlet that stores user preferences (e.g., theme color) in cookies and retrieves them on subsequent visits.	
5.	Create a JSP page to display the current date and time. Use expressions and scriptlets to implement logic.	
6.	Create a JSP page for user registration (name, email, password) and store the data in a database. Display stored users on another JSP page.	
7.	Create a Spring MVC application with a simple controller that returns a welcome message on the browser.	
8.	Create a registration form using Spring MVC. Use @ModelAttribute to capture user input and display confirmation.	
9.	Create a Spring application where a UserService class depends on UserRepository. Configure dependencies using XML and annotation-based DI.	
10	Create a simple Spring Boot application with REST endpoints to manage users (GET, POST). Test APIs using Postman.	
Asgn. No.	**Suggested List of Assignments**	
1	Understanding QA vs QC & Software Reliability Metrics Objective: Differentiate QA and QC processes and compute reliability metrics. Tasks: Define QA vs QC with examples from real software development. Calculate reliability metrics (MTTF, MTTR, MTBF, Availability) for a sample software system based on given data. Explore ISO 9000 quality standards and map them to software QA practices.	

2	Defects and Software Testing Lifecycle (STLC) Objective: Study defect terminology and map the Software Testing Life Cycle phases. Tasks: Identify and explain: Error, Fault, Bug, Defect, and Failure using real or hypothetical software issues. Map and explain STLC phases with examples. Write the difference between Verification vs Validation, and explain V-Model and W-Model with diagrams.	PART B
3	Writing Test Cases (Functional & Non-Functional Testing) Objective: Write functional (black-box) and non-functional test cases. Tasks: Choose a login module or calculator app. Write positive and negative test cases for each function. Write test cases for non-functional aspects (performance, usability, security). Document using IEEE 829 format (Test Plan, Test Case, Test Log).	
4	Static Testing – Reviews and Inspections Objective: Perform static testing using code/document reviews. Tasks: Conduct a code walkthrough or peer review for a small code snippet. Identify logical, syntactic, and standards-based issues. Fill out a review checklist and submit a review report. Use any static analysis tool (e.g., SonarLint or ESLint).	
5	Test Management Using JIRA Objective: Use JIRA for project and defect tracking. Tasks: Create a project in JIRA. Create user stories, tasks, and sub-tasks. Log defects with descriptions, steps to reproduce, and attach screenshots. Assign severity, priority, and track the Defect Life Cycle.	
6	Unit Testing Using TestNG (Java) Objective: Write and execute unit tests using a test framework. Tasks: Choose a small code module (calculator, bank transaction). Write unit test cases using TestNG (Java). Include assertions for boundary values, edge cases. Generate test reports and interpret results.	
7	Automation with Selenium WebDriver Objective: Automate browser-based testing using Selenium. Tasks: Automate a login page test: open browser, input credentials, and verify result. Use XPath/CSS selectors. Perform Data-driven Testing using Excel or CSV input. Generate logs and screenshots for failures.	
8	API Testing with Postman Objective: Test REST APIs using Postman. Tasks: Test CRUD operations on a sample REST API (e.g., reqres.in). Validate HTTP status codes, headers, and JSON responses. Use Tests tab in Postman to write small JavaScript assertions. Create and run a Collection of requests.	

9	Performance Testing using Apache JMeter Objective: Conduct load testing on a web application or API. Tasks: Configure a test plan with Thread Group and HTTP Request. Add listeners like Summary Report, Graph Results, View Results Tree. Run tests with varying users (10, 50, 100) and analyze throughput & response times. Optional: Record actions using JMeter proxy.
10	Building an Automation Framework (Mini Project) Objective: Create a simple Hybrid or Data-driven framework. Tasks: Combine Selenium (UI Testing), TestNG (Execution), Apache POI (Excel Input). Organize test scripts, utilities, test data, and reports into folders. Run tests with different data sets using the framework. Document structure and workflow of the framework.

Text Books:

1. Java Complete Reference Schildt Herbert, TMH.
2. Java Fundamentals (SIE), Schildt Herbert, TMH
3. The Complete Reference JSP, Phil Hanna, TMH
4. JDBC, Servlet and JSP, Black Book, Santosh Kumar K. Dremtech publication
5. Patton, R., Software Testing, BPB Publications, 2nd Edition, 2005
6. Kaner, C., Bach, J., Pettichord, B., Lessons Learned in Software Testing, Wiley, 1st Edition, 2002

Reference Books:

1. Head First Servlets and JSP, 2nd Edition by Bert Bates, Bryan Basham, Kathy Sierra
2. OCJP Oracle Certified Programmer for Java Study Guide by Kathy Sierra and Bert Bates.
3. A Programmer's Guide to Java OCJP Certification (A Comprehensive Primer) by Khalid A. Mughal and Rolf W. Rasmussen.
4. Java Server Programming Java Ee&(J2EE 1.7), Black Book, Wiley publications

E-Resources:

1. commended Learning Material
2. https://docs.oracle.com/en/java/javase/index.html
3. www.nptelvideos.com
4. https://www.geeksforgeeks.org/courses/search?query=java