Khandesh College Education Society's

Institute of Management and Research, Jalgaon

An Autonomous Institute affiliated
to

Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon

NEP-2020 Based CBCS

Program Structure and Syllabus

of

MASTER OF COMPUTER APPLICATIONS (MCA)

(with effect from 2024-25)

TABLE OF CONTENT

ABOUT THE PROGRAMME

PROGRAMME STRUCTURE & CREDIT DISTRIBUTION

Vision

To become a preferred Institute in the field of Computer Science and Technology, imparting relevant skills and knowledge to the students to meet emerging global challenges.

Mission

1. To impart up-to-date technical knowledge.
2. To develop students into successful professionals with analytical, logical and computational skills to provide sustainable solutions.
3. To expose students to the current trends and opportunities in local as well as global IT Industry.
4. To inculcate professional responsibility with an inherent ethical value system.
5. To cultivate innovative thinking and research culture among students.

Objective of the Programme

1. Practical Knowledge - Use Modern tools and technologies for software development.
2. Design and Development - Develop software solutions to problems across a broad range of application domains through analysis and design Contribute to research of their chosen field and function and communicate effectively, to perform both individually and in a multi-disciplinary team.
3. Innovation and Entrepreneurship - Continue the process of life-long learning through professional activities; adapt themselves with ease, to new technologies, while exhibiting high ethical and professional standards.

Name of the Programme

Master of Computer Applications
Duration of Program: 2 Years Full Time (4 Semesters)
Medium of Instruction: English

Description of the Programme

The Master of Computer Applications (MCA) program is designed to provide students with a comprehensive understanding of the field of computer science and its applications in various industries. MCA program incorporates the recently implemented National Education Policy (NEP) of 2020, which aims to transform the Indian education system and promote holistic development among students.

- MCA program is structured to equip students with the necessary knowledge and skills in computer science, programming, software development, and information technology.

It offers a blend of theoretical concepts and practical training, enabling students to apply their learning to real-world scenarios.

- The Programme will be of 2 years' duration with exit and entry options after 1 year. Students of this Programme can exit after 1st year with a PG Diploma, after 2nd year with a Post Graduate Degree.
- Students will be given opportunities for multidisciplinary and interdisciplinary education through options to choose courses of their interests from other schools/departments within the institute.
- The total credits for 2-year MCA will be **96** credits.
- 20% of the courses may be offered online from SWAYAM.
- Academic Bank of Credits (ABC) will be established to facilitate Transfer of Credits. The credits earned at various levels will get credited into a digitalized ABC. Students can use their earned credits to take admission in another institution to further continue their studies for the remaining year/s of their graduation.
- The Academic Calendar for this Programme of the institute will be synchronized to allow students of a particular PG Programme to study a course or courses from another PG Programme to meet the credit requirement of a semester. The commencement and closure of semesters and examinations for PG Programme will be planned in a uniform manner for declaration of results and awarding grades after a semester/year.

The Programme Highlights

Program Highlights: Master of Computer Applications (MCA) Program:

- **Discipline-Specific Courses (DSC) (Core Major Courses):** The MCA program places a strong emphasis on core major courses that form the foundationof computer science and applications. These courses provide in-depth knowledge and understanding of essential subjects such as programminglanguages, database management, software engineering, web development, data structures, algorithms, and computer networks.
- **Discipline-Specific Electives (DSE):** To cater to individual interests and specialization within the field of computer applications, the MCA program offers 8 department electives. These elective courses allow students to delve deeper into specific areas of computer science, such as Artificial Intelligence, Web UI Development, Java Programming, or Data Analytics. Each elective includes 4 hours of practical sessions in

addition to 4 hours of classroom sessions.

- **Field Project (FP) / On the Job Training (OJT) / Research Project (RP):** A student is required to undergo and successfully complete this course under the guidance of supervisor / mentor assigned by the HEI. This course must be corresponding to the major. This course must be completed at the HEI where the student has taken admission and transfer of credit is not permissible for this type of course. The project and internship component consists of minimum 18 weeks training, ensuring students gain practical industry experience.
- **Research Methodology (RM):** The inclusion of research methodology in the MCA program, as per the NEP, aims to equip students with essential research skills, fostering critical thinking and analytical abilities. This subject promotes innovation by enabling students to systematically investigate problems and contribute to technological advancements. Additionally, it enhances employability in research-oriented roles and aligns with the goal of promoting holistic education in higher studies.

Pedagogy for MCA Program:

The Master of Computer Applications (MCA) program adopts a student-centered and practical approach to learning, ensuring that students actively engage in the learning process and develop a strong foundation in computer science and applications. The pedagogy is designed to be simple yet effective, promoting holistic development and preparing students for successful careers in the field of Computer Applications.

- **Interactive Classroom Sessions**: The program fosters interactive classroom sessions where students actively participate in discussions, ask questions, and engage in problem-solving exercises. The faculty encourages student involvement and creates a supportive learning environment.
- **Demonstrations & Hands-on Lab Sessions**: Teaching is aided with practical demonstration of concepts to enhance the earning process. Practical sessions in well-equipped computer labs are an integral part of the MCA program. Students get hands-on experience with programming languages, software development tools, and other technologies. Lab exercises and projects allow them to apply theoretical concepts and gain practicalskills.
- **Real-world Examples**: The pedagogy includes the use of casestudies and real-world examples to demonstrate the application of concepts. By analyzing real-life scenarios

and exploring practical solutions, students develop critical thinking and problem-solving skills.

- **Industry Interaction**: The program encourages industry interaction through guest lectures, workshops, and industry visits. Professionals from the IT industry sharetheir experiences, insights, and current trends, giving students a glimpse into thepractical aspects of the field.
- **Project-based Learning**: The MCA program incorporates project-based learning, where students work on individual or group projects that simulate real-world scenarios. This approach enhances their teamwork, communication, and project management abilities while applying their knowledge to solve complex problems.
- **Internships and Practical Training**: The MCA program emphasizes internshipsand practical training opportunities. Students have the chance to work with industry partners, gaining hands-on experience, and applying their skills in real work environments. This exposure enhances their understanding of industry practices and prepares them for future employment.
- **Continuous Assessments**: Regular assessments, including quizzes, assignments,and presentations, help evaluate students' progress and understanding of the subject matter. Feedback is provided to guide their learning and address any gaps in understanding.
- **Technology Integration**: The program leverages technology as a learning tool. Online resources, educational software, and virtual labs are utilized to enhance students' understanding of concepts and provide additional learning opportunities.
- **Mentoring and Guidance**: Faculty members act as mentors, providing individual guidance and support to students. They assist in setting academic goals, clarifying doubts, and offering career advice to ensure students' overall growth and success.
- **Collaborative-Peer Learning**: The MCA program promotes collaborative learning through group projects, discussions, and peer-to-peer interactions. Students learnfrom each other, exchange ideas, and develop teamwork and communication skills.

The pedagogy of the MCA program aims to create a dynamic and engaging learningenvironment, enabling students to acquire theoretical knowledge, practical skills, and a problem-solving mindset. By incorporating these simple yet effective teaching strategies, the program equips students with the necessary competencies to thrive in the field of computer applications.

Two Years Master of Computer Applications Programme

The 2-year MCA PG degree will be of 96 Credits. Following types of courses will be offered for a 2-Year MCA Programme:

- 14 Discipline-specific Major Courses (44 credits)
- 8 Discipline Specific Electives (24 credits)
- 1 Research Methodology Course (4 credits)
- 1 On Job Training (12 credits)
- 1 Research / Minor Project (6 credits)
- 1 MOOC's Course (6 Credits)

Outcome Based Approach to Education (OBE):

As per the National Higher Education Qualification Frameworks (NHEQF), students are expected to possess the quality & characteristics of the graduate of a Programme of the study, including learning outcomes relating to the disciplinary areas, learning generic outcomes that are expected to be acquired by a graduate on completion of the Programme.

OBE is an educational model that forms the base of a quality education system. There is no specified style of teaching or assessment in OBE. All educational activities carried out in OBE should help the students to achieve the set goals. The faculty may adapt the role of an instructor, trainer, facilitator, and/or mentor based on the outcomes targeted. OBE enhances the traditional methods and focuses on what the institute provides to the students. It shows the success by making or demonstrating outcomes using statements 'able to do' in favor of students. It provides clear standards for observable and measurable outcomes.

Four Levels of Outcomes from OBE

1. Programme Educational Objectives (PEOs)
2. Programme Outcomes (POs)
3. Programme Specific Outcomes (PSOs)
4. Course Outcomes (COs)

Graduate Attributes

The graduate attributes include the learning outcomes that are specific to disciplinaryareas relating to the chosen field(s) of learning within the broad multidisciplinary & interdisciplinary learning outcomes that graduates of all Programs should acquire & demonstrate.

Graduate Attributes	
1.	Disciplinary Knowledge
2.	Critical Thinking & Problem Solving
3.	Creativity & Innovation
4.	Effective Communication
5.	Research-related skills
6.	Cooperation & Team Work
7.	Global/Multicultural Competence
8.	Ethics & Human Values
9.	Lifelong Learning
10.	Leadership Readiness
11.	Community Engagement & Social Responsibilities
12.	Digital literacy

Programme Educational Objectives (PEOs):

Programme Educational Objectives (PEOs) are defined for the aspiring students about what they will achieve once they join the Programme. PEOs are about professional and career enhancement after 2 years of post-graduation. PEOs are the written statements taken from different aspects like Knowledge, Skills & Ethics with focus on Career, Competency and Behavior. Three PEOs are recommended for MCA Programme.

Program Educational Objectives (PEOs):	
PEO1.	Use Modern tools and technologies for software development.
PEO2.	Develop software solutions to problems across a broad range of application domains through analysis and design.Contribute to research of their chosen fiel and function and communicate effectively, to perform both individually and in multi-disciplinary team.
PEO3.	Continue the process of life-long learning through professional activities adapt themselves with ease to new technologies, while exhibiting high ethical and professional standards.

Programme Outcomes (POs):

A Programme outcome is broad in scope and defines what the students will be able to do at the end of the Programme. POs are defined in line with the graduate attributes as specified above. POs are to be specific, measurable and achievable.

Programme Outcomes (POs):	
PO$_1$	Understand and apply mathematical foundation, computing and domain knowledge for the conceptualization of computing models from defined problems.
PO$_2$	Ability to identify, critically analyze and formulate complex computing problems using fundamentals of computer science and application domains.
PO$_3$	Ability to transform complex business scenarios and contemporary issues into problems, investigate, understand and propose solutions using emerging technologies.
PO$_4$	Ability to devise and conduct experiments, interpret data and provide well informed conclusions.
PO$_5$	Ability to select modern computing tools, skills and techniques necessary for innovative software solutions
PO$_6$	Ability to apply and commit professional ethics and cyber regulations in a global economic environment.
PO$_7$	Recognize the need for and develop the ability to engage in continuous learning as a Computing professional.
PO$_8$	Ability to understand, management and computing principles with computing knowledge to manage projects in multidisciplinary environments.
PO$_9$	Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations.
PO$_{10}$	Ability to recognize economic, environmental, social, health, legal, ethical issues involved in the use of computer technology and other consequential responsibilities relevant to professional practice.
PO$_{11}$	Ability to work as a member or leader in diverse teams in a multidisciplinary environment.
PO$_{12}$	Identify opportunities, entrepreneurship vision and use of innovative ideas to create value and wealth for the betterment of the individual and society.

Program Specific Outcomes (PSOs):

Programme Specific Outcomes (PSOs) are statements that describe what the Post Graduates of a specific Programme should be able to do. A list of 3 PSOs have been defined for the MCA Programme.

Program Specific Outcomes (PSOs)	
PSO$_1$.	Professionally skilled and trained in the field of computer science, theycan solve complex, real-time problems, which help them grow personally and professionally.
PSO$_2$.	Understanding modern computer technologies and their applications tosolve complex and critical issues that benefit society and the environment.
PSO$_3$.	Trained to perform effectively as an individual, a team, and as a teamleader in a multidisciplinary environment using critical thinking skills.

Mapping of PEOs with POs:

PEO	PO_1	PO_2	PO_3	PO_4	PO_5	PO_6	PO_7	PO_8	PO_9	PO_{10}	PO_{11}	PO_{12}
PEO_1	3	3	3	2	3	2	2	2	2	2	2	2
PEO_2	3	3	3	3	3	2	2	2	2	2	2	2
PEO_3	2	2	2	2	2	3	3	3	3	3	3	3
Level of correlation: 3-High, 2-Medium, 1-Low												

General Course Structure & Theme

A. Definition of Credit:

1 Hr. Lecture (L) per week	1 Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. Practical (P) per week	0.5 Credit
2 Hrs. Practical (P) per week	1 Credit

B. Course code and definition:

Course code	Definitions
L	Lecture
T	Tutorial
P	Practical
DSC	Discipline Specific Core Course
DSE	Discipline Specific Elective
OJT / FP	On Job Training: Internship / Apprenticeship / Field Project
RM	Research methodology
RP	Research Project
MOOCs	Massive Open Online Courses

C. Category wise Credits distribution:

Following types of courses will be offered for this 2 Years MCA

Type	Option 1 / Option 2	
	No. of courses	**Credits**
Discipline-Specific Courses (Core Major)	14	44
Discipline Specific Electives	8	24
Research Methodology	1	4
Skills Enhancement Courses (MOOCs)	3	6
On Job Training / Field Project	1	12
Research Project	1	6
Total		**96**

Course Level / Duration/System: Post graduate One or Two years / 2 or 4 Semesters with multiple entry and exit. The following option will be made available to the students joining MCA Programme:

a. One year: Post Graduate Diploma in Computer Applications.

b. Two years: Master of Computer Applications.

Note: Students can take extra credit course from their own department or from other department as per the Admitting Body / University n orms.

Year and Semester Wise Credits distribution

Years	Level	Semester	Major(Core) Subjects		RM	OJT, FP, RP	Cum. Cr/ Semester	Degree/ Cumulative Credit
			Mandatory (DSC)	**Elective (DSE)**				
I	6.0	I	20	06	---	---	26	52
		II	16	06	04	---	26	
		Cum. Cr.	36	12	04	---	52	
Credits after Ist Year			**36**	**12**	**04**	**---**	**52**	**52**
II	6.5	III	08	12		06	26	44
		IV	---	MOOCS 6	---	12	18	
		Cum. Cr.	08	18	---	12	44	
Credits after IInd Year			**44**	**30**	**04**	**18**	**96**	**96**

Duration of Programme	Total Credits	Degree
1 Years Program	52	PG Diploma in Computer Applications
2 Years Program	96	Master of Computer Applications

Proposed Syllabus Structure

KCES's Institute of Management and Research, Jalgaon

An Autonomous Institute, Affiliated to KBC, North Maharashtra University, Jalgaon

Course: MCA

Academic Year: 2024-25

Year	Sem	Type	Course Code	Title	L	P	Credit	Marks
First Year, SEMESTER – I, Level – 6.0								
I	I	DSC	MCA-DSC-511	Mathematical Foundations of Computer Science	4	--	4	100
			MCA-DSC-512	Computer Organization & Architecture	4	--	4	100
			MCA-DSC-513	Python Programming	4	--	4	100
			MCA-DSC-514	Database Management System	4	--	4	100
			MCA-DSC-515	Lab on Python Programming	--	2	2	50
			MCA-DSC-516	Lab on Database Management System	--	2	2	50
		DSE	MCA-DSE-517(A)	Basics of Web Development	4	--	4	100
			MCA-DSE-517 (B)	Java Programming				
			MCA-DSE-518 (A)	Lab on Basics of Web Development	--	2	2	50
			MCA-DSE-518 (B)	Lab on Java Programming				
				Total Credits	**20**	**06**	**26**	**650**
First Year, SEMESTER – II, Level – 6.0								
I	II	DSC	MCA-DSC-521	Computer Networks	4	--	4	100
			MCA-DSC-522	Operating System Concepts	4	--	4	100
			MCA-DSC-523	Data Structures & Algorithms	4	--	4	100
			MCA-DSC-524	Lab on Linux Operating System	--	2	2	50
			MCA-DSC-525	Lab on Data Structures & Algorithms	--	2	2	50
		DSE	MCA-DSE-526 (A)	Advance Web Development-I	4	--	4	100
			MCA-DSE-526 (B)	Advance Java Programming				
			MCA-DSE-526 (C)	Machine Learning				
			MCA-DSE-527 (A)	Lab on Advance Web Development-I	--	2	2	50
			MCA-DSE-527 (B)	Lab on Advance Java Programming				
			MCA-DSE-527 (C)	Lab on Machine Learning				
		RM	MCA RM 528	RM – Research Methodology	4	--	4	100
				Total Credits	**20**	**06**	**26**	**650**

Total Credit : 52 | Total Marks : 1300

Year	Sem	Type	Course Code	Title	L	P	Credit	Marks
Second Year, SEMESTER – III, Level – 6.5								
II	III	DSC	MCA-DSC-631	Fundamentals of Cloud Technologies	2	--	2	50
			MCA-DSC-632	Software Engineering	4	--	4	100
			MCA-DSC-633	Lab on Software Engineering	--	2	2	50
		DSE	MCA-DSE-634 (A)	Advance Web Development-II	4	--	4	100
			MCA-DSE-634 (B)	Adv. Artificial Intelligence				
			MCA-DSE-635 (A)	Web Development using Java	4	--	4	100
			MCA-DSE-635 (B)	Data Analytics				
			MCA-DSE-636 (A)	Lab on Advance Web Development-II	--	2	2	50
			MCA-DSE-636 (B)	Lab on Web Development using Java				
			MCA-DSE-637 (A)	Lab on Data Analytics		2	2	50
			MCA-DSE-637 (B)	Lab on Artificial Intelligence				
		RP	MCA-RP-638	Minor Project (Research / S/W Development)	--	6	6	150
				Total Credits	14	12	**26**	**650**
Second Year, SEMESTER – IV, Level – 6.5								
		OJT	MCA-OJT-641	Industrial Training	--	12	12	300
		MOOCS	MCA-MOOC-642 (A/B/C)	MOOCs - Online Certificate Courses	--	6	6	150
				Total Credits		18	**18**	**450**
Total Credit : 96 | Total Marks : 2400								

PROGRAM STRUCTURE AND CREDIT DISTRIBUTION
Master of Computer Application (MCA)

Year 1st , Sem-I, Level – 6.0					
Verticals	**Course Code**	**Subject**	**Theory / Practical**	**Credit**	**Marks**
Mandatory (DSC)	MCA-DSC-511	Mathematical Foundations of Computer Science	T	4	100
	MCA-DSC-512	Computer Organization & Architecture	T	4	100
	MCA-DSC-513	Python Programming	T	4	100
	MCA-DSC-514	Database Management System	T	4	100
	MCA-DSC-515	Lab on Python Programming	P	2	50
	MCA-DSC-516	Lab on Database Management System	P	2	50
Elective (DSE)	MCA-DSE-517(A)	Basics of Web Development	T	4	100
	MCA-DSE-517(B)	Java Programming			
	MCA-DSE-518 (A)	Lab on Java Programming	P	2	50
	MCA-DSE-518 (B)	Lab on Basics of Web Development			
Cumulative credits / Sem				**26**	**650**
Year 1st, Sem – II, Level – 6.0					
Mandatory (DSC)	MCA-DSC-521	Computer Networks	T	4	100
	MCA-DSC-522	Operating System Concepts	T	4	100
	MCA-DSC-523	Data Structures & Algorithms	T	4	100
	MCA-DSC-524	Lab on Linux Operating System	P	2	50
	MCA-DSC-525	Lab on Data Structures & Algorithms	P	2	50
Elective (DSE)	MCA-DSE-526 (A)	Advance Web Development-I	T	4	100
	MCA-DSE-526 (B)	Advance Java Programming			
	MCA-DSE-526 (C)	Machine Learning			
	MCA-DSE-527 (A)	Lab on Advance Web Development-I	P	2	50
	MCA-DSE-527 (B)	Lab on Advance Java Programming			
	MCA-DSE-527 (C)	Lab on Machine Learning			
RM	MCA RM 528	RM – Research Methodology	T	4	100
Cumulative credits / Sem				**26**	**650**
Cumulative Credits for MCA I				**52**	**1300**
Exit option: PG Diploma in Computer Applications (after 3 Years UG Degree)					

Year 2st , Sem-III, Level – 6.0

Verticals	Course Code	Subject	Theory / Practical	Credit	Marks
Mandatory (DSC)	MCA-DSC-631	Fundamentals of Cloud Technologies	T	2	50
	MCA-DSC-632	Software Engineering	T	4	100
	MCA-DSC-633	Lab on Software Engineering	P	2	50
Elective (DSE)	MCA-DSE-634 (A)	Advance Web Development-II	T	4	100
	MCA-DSE-634 (B)	Adv. Artificial Intelligence			
	MCA-DSE-635 (A)	Web Development using Java	T	4	100
	MCA-DSE-635 (B)	Data Analytics			
	MCA-DSE-636 (A)	Lab on Advance Web Development-II	P	2	50
	MCA-DSE-636 (B)	Lab on Web Development using Java			
	MCA-DSE-637 (A)	Lab on Data Analytics	P	2	50
	MCA-DSE-637 (B)	Lab on Artificial Intelligence			
Research Project (RP)	MCA-RP-638	Minor Project (Research / S/W Development)	--	6	150
Cumulative credits / Sem				**26**	**650**
First Year, SEMESTER – II, Level – 6.0					
On Job Training (OJT)	MCA-OJT-641	Industrial Training	--	12	300
MOOCs	MCA-MOOC-642 (A/B/C)	MOOCs - Online Certificate Courses	--	6	150
Cumulative credits / Sem				**18**	**450**
Cumulative Credits for MCA II				**44**	**1100**
2 Years 4 Semesters MCA Degree : Total Credit : 96 | Total Marks : 2400					

Semester-I

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

MCA-DSC-511: Mathematical Foundations of Computer Science

Course Title: Mathematical Foundations of Computer Science	Course Type: DSC
Course Code: MCA-DSC-511	Total Credits: 04
Lectures: Tutorials: Practical: 4:0:0	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course provides a comprehensive introduction to the mathematical and statistical principles that underpin computer science. It is designed to equip students with the necessary tools to understand, analyse computational methods and algorithms.

Course Objectives:

1. To build the foundation of computer algorithms using mathematical base.
2. To apply statistical measures on the data and represent it graphically.
3. To relate practical examples to the probability theory and probability distributions to build the foundation for machine learning.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment
✓	✓			✓	✓

Course Outcomes: At the end of the Course, the Student will be able to:

CO1	**Understand** the permutation, combination and pigeonhole principle to solve real time problems.
CO2	Able to formulate and **solve** recurrence relations problems.
CO3	**Analyse** the behaviour of the data, model the data using statistical measures and represent it graphically on paper without using available computerized tools.
CO4	**Analyse** the basic concepts of probability theory.
CO5	**Analyse** the basic concepts of probability distribution, mass functions and Stochastic Processes.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	1	1	2	1	1	1	1	1	1	1
CO2	2	3	1	1	2	1	1	1	1	1	1	1
CO3	1	2	1	2	1	1	1	1	1	1	1	1
CO4	1	2	1	2	1	1	1	1	1	1	1	1
CO5	1	2	1	2	1	1	1	1	1	1	1	1

SN	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Unit 1 Induction and Recursion:** Mathematical Induction, Strong Induction and Well Ordering, Recursive Algorithms, Program Correctness, The Basics of Counting, The Pigeonhole Principle, Permutations and Combinations.	10	12	**CO1**
2	**Unit 2 Advance Counting Techniques:** Recursive Relations, The Towers of Hanoi, Merge Sort, Linear Recurrences, Solving Linear Recurrence Relations, Divide-and-Conquer Recurrences, Divide-and-Conquer Algorithms, Generating Functions, Inclusion-Exclusion, Applications of Inclusion Exclusion.	10	16	**CO2**
3	**Unit 3 Statistics:** Population, sample, parameters, and statistics: definition, methods of sampling, types of variables, applications, Data Presentation: Classification of data, Frequency distribution, Cumulative and Relative frequency distribution, Descriptive Statistics: Central tendency-mean, median, mode, range, quartile deviation, , variance, standard deviation ,Graphical representation of statistical data	10	16	**CO3**
4	**Unit 4 Probability:** Making decisions under uncertainty, Classical definition of Probability, Events and their Outcomes, Rules of Probability, Probability axioms, Joint and Conditional probability, Probability independence, and Bayes theorem,	10	16	**CO4**
5	**Unit 5 Probability Distributions:** Random variables (discrete and continuous), Probability mass function, Distributions: Binomial, Poisson, Probability density function, Distributions: Uniform, Exponential, Normal.	10	15	**CO5**
6	**Unit 6 Stochastic Processes:** Definitions and classifications of Stochastic Processes, discrete and continuous Markov models, Chapman-Kolmogorov equation.	10	15	**CO5**

Reference Books:

1. Kenneth H. Rosen, Discrete Mathematics, and its Applications 6th Ed, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2007 ISBN 10: 0070681880
2. Michael Baron (2014) Probability and Statistics for Computer Scientists Second Edition, CRC press. ISBN: 978-1-4822-1410-9
3. Goon A.M., Gupta M.K., Dasgupta. B. (2001), Fundamentals of Statistics, Volume I and II, World Press, Calcutta.
4. Ross, S. (2005). Introduction to Probability Models, (6th Ed. Academic Press). ISBN 978 25 0-12-375686-2
5. Medhi, J. (1994). Stochastic Processes, (2nd Ed. New Age Publisher) ISBN : 978-93-86286 48-2

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	
End Semester Examination (60)	✓	✓	✓	✓	✓	

SEMESTER: I

MCA-DSC-512: Computer Organization & Architecture.

Course Title: Computer Organization & Architecture.	Course Type: DSC
Course Code: MCA-DSC-512	Total Credits: 04
Lectures: Tutorials: Practical: 4:0	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

The computer lies at the heart of computing. All students of computing should acquire some understanding and appreciation of a computer system's functional components, their characteristics, their performance, and their interactions.

It is important to understand Computer Architecture in order to structure a program so that it runs efficiently on a real machine. And when selecting a system to use, it is important to understand the tradeoff among various components to accurately compare competing systems, and understand technical literature on new computer systems.

This course will cover the basic concepts of Computer Architecture that are important for students to understand, including the digital subsystem, CPU control unit and datapath, memory systems including caching and virtual memory, and input/output subsystems.

Course Objectives:

1. To Understand the fundamental organization of a computer system.
2. To understand various number systems, different methods used for the simplification of Boolean functions.
3. To design and implement a system that uses combinational logic for the given specification.
4. To design and implement synchronous sequential system for the given specification.
5. To Understand the basics of instructions sets and their impact on processor design.
6. To Understand the addressing modes, instruction formats and program control statements.
7. To Understand the I/O interface, I/O operations and memory system.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to:

CO1	Describe the fundamental organization of a computer system.
CO2	Perform arithmetic operations in any number system Simplify the Boolean expression using K –Map and Tabulation techniques
CO3	Use Boolean Simplification techniques to design a combinational hardware circuit Design and analysis of a given digital Combinational circuit and sequential circuits
CO4	Understand the basics of instructions sets and their impact on processor design.
CO5	Understanding of the addressing modes, instruction formats and program control statements.
CO6	Understand the I/O interface, I/O operations and memory system.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	1	-	-	-	-	-	-	-
CO2	3	2	1	2	2	-	-	-	-	-	-	-
CO3	2	3	2	2	1	-	-	-	-	-	-	-
CO4	3	2	-	-	2	-	-	-	-	-	-	-
CO5	2	2	2	-	2	-	-	-	-	-	-	-
CO6	2	1	2	1	2	-	-	-	-	-	-	-

SN	Course Contents	Hrs	Marks
1	**INTRODUCTION TO COMPUTERS ARCHITECTURE, BOOLEAN ALGEBRA AND LOGIC GATES** Basic of Computers, Computer Architecture, Von Neumann Architecture, Generation of Computers, Classification of Computers. Review of Number Systems – Arithmetic Operations – Binary Codes – Boolean Algebra and Theorems – Boolean Functions – Simplification of Boolean Functions using Karnaugh Map and Tabulation Methods – Logic Gates – NAND and NOR Implementations.	12	18
2	**COMBINATIONAL CIRCUITS** Combinational Circuits – Analysis and Design Procedures – Half Adder, Full Adder, Code Conversion – Decoders and Encoders – Multiplexers and De multiplexers.	8	12
3	**SEQUENTIAL CIRCUITS** Sequential Circuits – Latches and Flip Flops – Analysis and Design Procedures – State Reduction and State Assignment – Shift Registers – Counters.	10	18--

SN	Course Contents	Hrs	Marks
4	**DATA REPRESENTATION AND ARITHMETIC** Signed Magnitude: 1’s Complements, 2’s Complements, Floating Point data, other Representation: BCD, Gray Code, Arithmetic operations: Addition, Subtraction, Multiplication, and Division.	9	12
5	**PROCESSOR ORGANIZATION** General Register Organization - ALU - Instruction codes Instruction Formats- Instruction sets - Stack Organization - Addressing modes.	8	12
6	**CONTROL UNIT** Register transfer, Bus and memory transfer, micro operations, micro programming, hardwired control, RISC, CISC.	6	6
7	**INPUT/OUTPUT AND MEMORY ORGANIZATION** I/O interface, Asynchronous data transfer, Modes of transfer, priority Interrupt, Direct memory access, Memory Hierarchy, Main memory, Auxiliary memory, Associate Memory, Cache Memory, and Virtual memory.	7	12

REFERENCES:

1. Morris Mano M. and Michael D. Ciletti, “Digital Design”, IV Edition, Pearson Education, 2008.
2. John F. Wakerly, “Digital Design Principles and Practices”, Fourth Edition, Pearson Education,2007.
3. M. Morris Mano, (l992),Computer System Architecture, Eastern Economy Edition (Third Edition), Prentice Hall of India Pvt. Ltd, ISBN: 9780131755635
4. WilliamStallings , Computer Organization and Architecture Describing for Performance, Eastern Economy Edition. (Fourth Edition),ISBN: 13: 9780136073734.
5. John. P. Hayes, (1998),Computer System Architecture, 3rd edition, Prentice Hall of India Pvt. Ltd, ISBN: 0071159975
6. Hwang K. Briggs, (1984), Computer Architecture and parallel Processing,3rd edition, McGraw- Hill, ISBN:0070315566.

Assessment Pattern:

Bloom’s Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓		✓	
End Semester Examination (60)	✓	✓	✓		✓	

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

MCA-DSC-513: Python Programming

Course Title: Python Programming	Course Type: DSC
Course Code: MCA-DSC-513	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 90

Course Description:

This course contains core concepts of the Python programming language, focusing on both foundational and advanced concepts. Students will learn to write efficient code, work with data structures, implement object-oriented programming, and handle files and exceptions. The course also covers modules, libraries, and database connectivity, preparing students to develop robust Python applications

Course Objectives:

1. Understand the fundamentals of Python programming and its syntax.
2. Develop the ability to write Python programs for a variety of applications.
3. Master Python data structures, functions, and modules to create efficient and reusable code.
4. Implement object-oriented programming principles in Python.
5. Gain proficiency in handling files, exceptions, and databases.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Write clear, efficient, and well-documented Python programs.
CO2	Use Python's data structures such as lists, dictionaries, and sets effectively.
CO3	Implement object-oriented programming concepts like classes, inheritance, and polymorphism in Python.
CO4	Handle file operations and exceptions gracefully in Python programs.
CO5	Connect to databases and perform CRUD operations using Python.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	3	2	1	1	3	1	1	1
CO2	3	3	2	2	3	1	2	1	2	1	1	1
CO3	3	3	3	2	3	2	2	2	2	1	2	2
CO4	2	2	2	3	3	1	2	1	2	2	1	1
CO5	3	3	3	2	3	2	2	2	2	2	2	2
CO6												

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Unit 1: Introduction to Python Programming** **1.1. Overview of Python** • History of Python • Python's Popularity and Applications • Installing Python and Setting Up the Environment **1.2. Python Basics** • Python Syntax and Semantics • Variables, Data Types, and Operators • Input/Output Operations • Writing and Executing Python Programs **1.3. Control Structures** • Conditional Statements (if, elif, else) • Looping Structures (for, while) • Control Flow Tools (break, continue, pass)	10	15	**CO1**
2	**Unit 2: Python Data Structures and Functions** **2.1. Lists, Tuples, and Dictionaries** • Lists: Creation, Manipulation, and Methods • Tuples: Immutable Sequences • Dictionaries: Key-Value Pairs and Methods • Sets and their Operations **2.2. Functions in Python** • Defining and Calling Functions • Function Arguments and Return Values • Lambda Functions and Anonymous Functions • Scope and Lifetime of Variables	10	15	**CO2, CO1**

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
3	**Unit 3: Object-Oriented Programming (OOP) in Python** **3.1. Introduction to OOP Concepts** • Understanding Classes and Objects • Defining and Using Methods • Constructors and Destructors **3.2. Advanced OOP Concepts** • Inheritance: Single, Multiple, and Multilevel • Polymorphism: Method Overloading and Overriding • Encapsulation and Data Hiding	10	15	**CO3**
4	**Unit 4: File Handling and Exception Handling** **4.1. File Operations in Python** • Opening, Reading, and Writing Files • Working with Text and Binary Files • File Methods and Context Managers (with statement) **4.2. Exception Handling** • Understanding Errors and Exceptions • Try, Except, Finally, and Else Clauses • Raising Exceptions • Custom Exception Classes	10	15	**CO4**
5	**Unit 5: Modules, Packages, and External Libraries** **5.1. Modules and Packages** • Creating and Using Modules • Importing Specific Attributes from Modules • Creating and Using Python Packages • The Python Standard Library Overview **5.2. Working with External Libraries** • Installing and Using External Libraries (pip) • NumPy-Array Operations, Mathematical Functions for numerical computations. • Pandas-Data Structures: Series and DataFrame, Data Manipulation: data cleaning, filtering, and analysis.	12	18	**CO1**
6	**Unit 6: Database Connectivity with Python** **6.1. Database Connectivity** • Introduction to Databases and SQL • Connecting Python to a Database (e.g., SQLite, MySQL) • Executing SQL Queries from Python • Handling Transactions and Database Operations	08	12	**CO5,** **CO1**

Reference Books:

1. "Python Crash Course" by Eric Matthes**,** published by No Starch Press in 2019 (2nd Edition), with ISBN 978-1593279288"Learning Python" by Mark Lutz
2. "Learning Python" by Mark Lutz**,** published by O'Reilly Media in 2013 (5th Edition), with ISBN 978-1449355739
3. "Python Programming: An Introduction to Computer Science" by John Zelle**,** published by Franklin, Beedle & Associates Inc. in 2016 (3rd Edition), with ISBN 978-1590282755

4. "Python for Everybody: Exploring Data in Python 3" by Charles Severance, published by Create Space Independent Publishing Platform in 2016, with ISBN 978-1530051120

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓			

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

MCA-DSC-514 Database Management System

Course Title: Database Management System	Course Type: DSC
Course Code: MCA-DSC-514	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course introduces the core principles and techniques required in the design and implementation of database systems. This course focus on relational database management systems, including database design theory: E-R modelling, data definition and manipulation languages, database security and administration. It also covers essential DBMS concepts such as: Transaction Processing, Concurrency Control and Recovery and various types of databases. It also provides students with knowledge of NoSQL and MongoDB.

Course Objectives:

At the end of the course, the students will be able to:

2. Understand the basic concepts and the applications of database systems.

3. Master the basics of SQL and construct queries using SQL.

4. Understand the relational database design principles.

5. Familiar with the basic issues of transaction processing and concurrency control.

6. Familiar with NoSQL and MongoDB.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –	
CO1	Understand the fundamental concepts and terminology associated with database systems, including architecture, design, and various models.
CO2	Design entity-relationship diagrams to the given problem to develop database application with appropriate fields and validations.
CO3	Implement a database schema for a given problem domain.
CO4	Formulate SQL queries to the given problem
CO5	Understand and Apply normalization techniques to improve the database design to the given problem and build database for any problem

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	1	-	-	-	-	-	-	-
CO2	3	2	2	1	1	-	-	-	-	-	-	-
CO3	3	2	-	2	2	-	-	-	-	-	-	2
CO4	3	2	2	2	2	-	-	-	-	-	-	2
CO5	3	2	-	-	2	-	-	-	-	-	-	2
CO6	3	3	2	2	2	-	-	-	-	-	-	2

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Introduction to DBMS** Characteristics of Database approach, Peoples associated with Database system. Advantages of using DBMS approach, A Brief History of Database Applications, Data models, schemas and instances Three-schema architecture and data independence, Database languages and interfaces The database system environment, Centralized and client-server architectures.	10	18	
2	**Entity-Relationship Model** Entity-Relationship Model: Conceptual Database using high level conceptual data models for Database Design, A Sample Database Application. Entity types, Entity sets Attributes and Keys Relationship types. Relationship Sets, Roles, and Structural Constraints, Weak entity types ER Diagrams, Naming Conventions.	12	18	
3	**Relational Algebra** Relational Algebra: Basic Relational Algebra operations. Set theoretical operations on relations. Relational algebra queries. JOIN operations Aggregate Functions and Grouping.	8	12	
4	**SQL and Aggregate Functions** SQL data definition and data types specifying constraints in SQL, basic retrieval queries in SQL Insert, update and delete statements in SQL aggregate functions in SQL. Group by and having clauses	14	18	
5	**Functional Dependencies and Transaction Processing** Basics of functional dependencies and normalization for relational databases. Informal design guidelines for relational schemas, functional dependencies. Normal forms – First normal form, Second normal form, Third normal form. Boyce-Codd normal form. Introduction to Transaction Processing. Single user & multiuser systems. Transactions: read & write operations. Need of concurrency control: The lost update problem, Dirty read problem. Types of failures. Transaction states. Desirable properties (ACID properties) of Transactions.	10	18	

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
6	**Introduction to NoSQL** NoSQL: What It Is and Why You Need It, Definition and Introduction Types of NoSQL databases. Hands on experience with NoSQL, First Impressions: A simple set of persistent preferences data Storing car make and model data, Working with Language Bindings: Mongo DB drivers	6	6	

Textbooks:

1. Fundamentals of Database Systems, Elmasri and Navathe, Pearson International Publications, 7th edition, 2015. Chapters: 1,2,3, 6, 14, 20.
2. Professional NoSQL, Shashank Tiwari, Wrox Publications, 2011. Chapters: 1 and 2

Reference Books:

1. Fundamentals of Database Systems, Mark L Gillenson, 2nd Edition, John Wiley & Sons, 2011.
2. NoSQL: Database for Storage and Retrieval of Data in Cloud, Ganesh Chandra Deka,

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation (40)	✓	✓	✓	✓	✓	
End Semester Examination (60)	✓	✓	✓	✓	✓	

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

MCA-DSC-515: Lab on Python Programming

Course Title: Lab on Python Programming	Course Type: DSC
Course Code: MCA-DSC-515	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. To understand and apply Python's core syntax, variables, data types, and control structures like loops and conditionals.
2. To enhance logical thinking and algorithm design by solving computational problems using Python.
3. To learn to manipulate data using Python's built-in structures (lists, dictionaries, etc.) and handle file input/output operations.
4. To gain foundational knowledge of object-oriented principles, including classes, objects and database to write more structured and scalable code.

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
x	✓	✓	✓	x	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	**Visualize** and execute Python code confidently, using appropriate syntax and programming constructs.
CO2	Design and **apply** algorithms and solve real-world problems using Python.
CO3	Manipulate and **analyze** data using Python's data structures and perform file operations.
CO4	**Develop** and demonstrate knowledge of object-oriented programming principles by creating and using classes, objects, and databases in Python programs.

Sr. No.	Assignments
1	**Simple Python programs:** a) Write and execute simple Python programs that demonstrate variable assignments, basic operations, and input/output functions.

2	**Implementing Control Structures** a) Write Python programs that use conditional statements (if, elif, else) and loops (for, while). b) Create a program to calculate the factorial of a number using a loop.
3	**Working with Lists, Tuples, and Dictionaries** a) Implement a Python program to demonstrate the creation, manipulation, and use of lists, tuples, and dictionaries.
4	**Creating and Using Functions** a) Write Python programs to define and call functions with different argument types (positional, keyword, default). b) Implement a program using anonymous functions.
5	**Creating Classes and Objects** a) Write a Python program to create a Student class with attributes like name, age, and marks. Create objects and demonstrate the use of methods.
6	**Implementing Inheritance and Polymorphism** a) Implement a program to demonstrate single and multilevel inheritance. b) Demonstrate method overloading and overriding with a real-world example.
7	**Performing File Operations** a) Write a Python program to read and write data to a text file.
8	**Implementing Exception Handling** a) Write Python programs to handle exceptions using try, except, finally, and else clauses.
9	**Creating and Using Python Modules** a) Write a Python program to create and use custom modules.
10	**Working with External Libraries** a) Install the NumPy Library and perform use of Random Numbers, Mathematical, Statistical, Linear Algebra operations, Sorting, Searching, and Indexing on NumPy arrays. b) Install the Pandas Library and apply filtering techniques to extract specific data and calculate basic statistics.
11	**Connecting Python to a Database** c) Write a Python program to connect to an SQLite database, create tables, insert data, and execute queries.

Tools: Compilers of any programming language (viz. C / C++ / Java / C# / Python)

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓	✓	✓	✓

End Semester Examination (30)	✓	✓	✓	✓	✓	✓

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

MCA-DSC-516: Lab on Database Management System

Course Title: Database Management System	Course Type: DSC
Course Code: MCA-DSC-516	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lecture Hours: 30 Hours	ESE Marks: 30

Course Description and Objective:

The objective of this lab course is to understand the practical applicability of database management system concepts. Working on existing database systems, designing of database, creating relational database, analysis of table design. The lab course also provide practical knowledge to understand NoSQL and MongoDB.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Students get practical knowledge on designing and creating relational database systems
CO2	Understand various advanced queries execution such as relational constraints, joins, set operations, aggregate functions.
CO3	Use of various software to design and build ER Diagrams, UML, Flow chart for related database systems.
CO4	Students will be able to design and implement database applications on their own.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	2	2	1	-	-	-	-	-	-	2
CO2	3	2	2	1	1	-	-	-	-	-	-	-
CO3	3	2	-	-	2	-	-	-	-	-	-	2
CO4	3	3	2	-	1	-	-	-	-	-	-	-

Sr. No.	*Assignments*
1	**E-R Model:** Analyze the problem carefully and Identify the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys **Concept design with E-R Model:** Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc wherever required.
2	**DDL and DML commands**: Practice of DDL commands, DML commands, Constraints and Data Query Language
3	**SQL Special operators**: Practice to Convert ER Diagrams into tables and SQL Special operators like, (in between, is null, not, exist, not Exists ANY, ALL, IN, set operators, Constraints etc.
4	**Aggregate functions**: To practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views. **Functions**: To practice queries on String/Character,Date/Time Functions,Numeric Functions,Math Functions
5	**SQL Joins:** To practice queries on different Joins in SQL. **Sub Queries**: To practice queries on Nested Queries, Correlated Sub Queries. **DCL commands** To practice queries on DCL Commands **Views**: To practice queries on Create views.
6	**NoSql:** Storing car make and model data, Working with Language Bindings: Mongo DB drivers.

Textbooks:

3. Fundamentals of Database Systems, Elmasri and Navathe, Pearson International Publications, 7th edition, 2015. Chapters: 1,2,3, 6, 14, 20.
4. Professional NoSQL, Shashank Tiwari, Wrox Publications, 2011. Chapters: 1 and 2

Reference Books:

3. Fundamentals of Database Systems, Mark L Gillenson, 2nd Edition, John Wiley & Sons, 2011.
4. NoSQL: Database for Storage and Retrieval of Data in Cloud, Ganesh Chandra Deka,

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓	✓	✓	✓

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

ELECTIVE COURSE

MCA-DSE-517(A): Basics of Web Development

Course Title: Basics of Web Development	Course Type: DSE
Course Code: MCA-DSE-517(A)	Total Credits: 04
Lectures: Tutorials: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course offers a comprehensive introduction to web development, focusing on the essential skills for creating and styling web pages and developing interactive applications. Students will learn HTML basics and advanced elements, CSS for styling and responsive design, and JavaScript for dynamic web interactions. Topics include client-server architecture, semantic HTML, CSS Grid and Flexbox, animations, and JavaScript DOM manipulation. By the end of the course, students will be equipped to build modern, responsive web applications.

Course Objectives:

1. Develop and structure web pages using HTML, incorporating both basic and advanced HTML elements.
2. Apply CSS to style web pages, utilizing fundamental concepts such as the box model, typography, and various CSS properties.
3. Design responsive web layouts using advanced CSS techniques, including CSS Grid, Flexbox, media queries, and animations.
4. Explore advanced CSS concepts, including pre-processors like SASS and the use of CSS variables and custom properties.
5. Write and implement JavaScript code to manipulate the DOM, manage events, and enhance the interactivity of web pages.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment
✓	✓	✓	✓	✓	✓

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	**Understand** the client-server architecture, HTML document structure, and process of rendering web pages over the Internet.
CO2	**Develop** well-structured web pages using HTML elements, forms, and multimedia, and apply CSS for layout, and responsive design.
CO3	**Utilize** CSS Grid, Flexbox, animations, transitions, and pre-processors like SASS to create responsive, visually appealing, and efficient web layouts.
CO4	**Apply** the various JavaScript features such as DOM, and events to **create** interactive web content and provide a dynamic user experience.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	--	--	--	2	--	2	--	2	--	--
CO2	--	3	3	--	3	--	--	--	3	--	--	3
CO3	--	--	3	--	3	--	--	--	--	--	3	--
CO4	3	--	--	3	--	--	3	--	--	--	--	--

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Unit-1: Introduction to Web Development and HTML Basics** 1.1 Overview of Web Development (Client-Server Architecture, Internet Basics). 1.2 HTML Syntax and Document Structure. 1.3 Basic HTML Tags: <html>, <head>, <title>, <body>, <h1> to <h6>, <p>, <a>, <img>, <ul>, <ol>, <li>, <div>, <span>. 1.4 Creating a Simple Webpage	8	10	CO1
2	**Unit-2: Advanced HTML** 2.1 Semantic HTML: New HTML5 Semantic Elements: <header>, <footer>, <article>, <section>, <nav>, <aside> 2.2 Forms and Input Elements: 2.2.1 Creating Forms (<form>, <input>, <textarea>, <button>, <select>, <option>) 2.2.2 Form Attributes (required, placeholder, disabled, etc.). 2.3 Multimedia Integration: Embedding Audio and Video (<audio>, <video>).	10	12	CO2
3	**Unit-3: CSS Fundamentals** 3.1 Introduction to CSS: 3.1.1 CSS Syntax and Selectors (element, class, id, attribute selectors). 3.1.2 CSS Properties (color, background, font, text alignment). 3.2 Box Model: Content, Padding, Border, Margin. Styling Borders, Margins, and Padding. 3.3 Text and Font Styling: Typography, Font Properties (font family, size, weight, style).	8	14	CO2

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
	3.4 CSS Units and Values: Absolute vs. Relative Units (px, em, rem, %, vh, vw).			
4	**Unit-4: Advanced CSS Techniques** 4.1 Responsive Design: 4.1.1 Media Queries and Breakpoints 4.1.2 Flexible Grid Layouts and Flexbox. 4.2 CSS Grid: 4.2.1 Introduction to CSS Grid Layout. 4.2.2 Creating Complex Layouts with CSS Grid. 4.3 CSS Animations and Transitions: 4.3.1 Basic Transitions and Animations.CSS Units and Values 4.3.2 Key-frames and Animation Properties	12	18	CO3
5	**Unit-5: Advanced CSS Techniques and Pre-processors** 5.1 Advanced Selectors and Combinatory: 5.1.1 Attribute Selectors, Child and Sibling Selectors 5.1.2 Pseudo-classes and Pseudo-elements 5.2 CSS Variables and Custom Properties: 5.2.1 Introduction to CSS Variables. 5.2.2 Introduction to CSS Variables. 5.3 CSS Pre-processors (SASS): 5.3.1 Introduction to SASS 5.3.2 Variables, Nesting and Inheritance. 5.3.3 Compiling and Using Pre-processed CSS.	10	16	CO3
6	**Unit-6: Introduction to JavaScript and DOM Manipulation** 6.1 Introduction to JavaScript: HTML Syntax and Document Structure. 6.1.1 JavaScript Syntax, Variables, Data Types. 6.1.2 Operators, Conditional Statements, Loops. 6.2 Functions and Events Creating a Simple Webpage 6.2.1 JavaScript Syntax, Variables, Data Types. 6.2.2 Operators, Conditional Statements, Loops. 6.3 DOM Manipulation: 6.2.3 Selecting and Modifying DOM Elements. 6.2.4 Creating, Deleting, & Updating Elements dynamically	12	20	CO4

Reference Books:

1. HTML & CSS: Design and Build Websites, by Jon Duckett, Wiley Publication, 1st Edition, ISBN-10: 1118008189, ISBN-13: 978-1118008188
2. CSS: The Definitive Guide, by Eric A. Meyer and Estelle Weyl, O'Reilly Media Publication, 4th Edition, ISBN-10: 1449393195, ISBN-13: 978-1449393199JavaScript:
3. The Definitive Guide, by David Flanagan, O'Reilly Media Publication, 7th Edition, ISBN-10: 1491952024, ISBN-13: 978-1491952023
4. Online Resources: MDN Web Docs, W3Schools, CSS-Tricks, and other tutorial sites.

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓			

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

ELECTIVE COURSE

MCA-DSE-517 (B): Java Programming

Course Title: Java Programming	Course Type: DSE
Course Code: MCA-DSE-517 (B)	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course covers the foundations of Java programming that mainly includes Core Java concepts with Object Oriented Programming Concepts.

Course Objectives:

7. Understand Fundamental concepts of object oriented programming using Java technology.
8. Developing simple Java applications using OOPs concepts.
9. Developing Java applications using abstract classes, inheritance and interfaces
10. Creating classes, functions, packages for writing Java Applications
11. Implementing Multithreading in small projects

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	State fundamental concepts of object oriented programming using Java technology.
CO2	Write class definition for different problems using OOPs concepts
CO3	Illustrate abstract classes, inheritance and interfaces in Java applications
CO4	Analyse the problem for designing classes, functions, packages for Java Applications
CO5	Apply generic programming for small projects using collection class, files and databases
CO6	Develop Small Java applications using required OOPs concepts of Core Java

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	1	1	1	1	2	2	1	1	1
CO2	2	2	2	2	3	3	3	2	2	2	1	1
CO3	2	2	2	1	1	1	2	2	2	1	1	1
CO4	3	3	3	2	3	3	3	2	2	1	1	2
CO5	2	2	2	2	2	2	1	1	1	1	2	2
CO6	2	2	1	3	2	2	2	2	2	3	3	1

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Fundamental Programming Structures** 1.1 Write First Program 1.2 Understanding Structure of Java Program 1.3 Primitive Types 1.3 Variables 1.4 Arithmetic Operations 1.5 Strings 1.6 Input and Output 1.7 Control Flow 1.8 Arrays and Array Lists, Matrices 1.9 Functional Decomposition	8	5	**CO1** **CO2**
2	**Object-Oriented Programming** 2.1 Working with Objects 2.2 Implementing Classes 2.3 Object Construction 2.4 Records 2.5 Static Variables and Methods 2.6 Abstraction, Encapsulation, Packages 2.7 Nested Classes (Inner Class) 2.8 Documentation Comments	8	10	**CO2**
3	**Interfaces and Lambda Expressions** 3.1 Abstract classes and Interfaces 3.2 Static, Default, and Private Methods 3.3 Examples of Interfaces 3.4 Lambda Expressions 3.5 Methods and Constructor References 3.6 Processing Lambda Expressions 3.7 Lambda Expressions and Variable Scope 3.8 Higher-Order Functions 3.9 Local and Anonymous Classes	10	10	**CO3**
4	**Inheritance and Reflection** 4.1 Extending a Class 4.2 Inheritance Hierarchies 4.3 Object: The Cosmic Superclass 4.4 Enumerations	10	10	**CO4**

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
	4.5 Runtime Type Information and Resources 4.6 Reflection			
5	**Multithreading and Exceptions** **5.1** Creating Thread 5.2 Multi-Tasking using Threads, Thread 5.3 Synchronization or Thread Safe, 5.4 Thread Class Methods, 5.5 Thread Communication, Thread, Properties, Thread Group, Thread States (Life-Cycle of a Thread) 5.6 Exception handling (try, catch, finally), throws clause, throw clause, 5.7 Types of Exceptions(built-in, user defined), Assertions and Logging	4	5	**CO4**
6	**Generic programming:** a. Generic Class, Generic Method, Generic Interface. b. Generic Class examples for Stack and Queue data structures **The Collection framework:** c. Collection Object, d. Retrieving elements from Collection, e. LinkedList Class, HashSet Class, LinkedHashSet Class f. ArrayList Class, Vector Class, HashMap Class, g. Arrays Class; Object Wrappers and Autoboxing	6	10	**CO5**
7	**Graphics Programming and User Interface** **7.1** Graphics class, Graphics2D class, 7.2 Drawing lines, circle, ellipses, rectangle, arcs and polygons. 7.3 Setting Colors-background, foreground 7.4 GUI designing using swing classes, 7.5 Event handling using listeners – action Listener, mouse Listener, and other	10	10	**CO6**

Reference Books:

1. Cay's Horstmann and Gary Cornell, (2012), Core Java Volume -1 Fundamentals, 12th Edition.
2. Horstman Cay, Cornell Gary, Core JavaTM2, Vol.1&2, 11th Edition, Pearson education.
3. E Balguruswamy, (2000), Programming in Java, Tata McGraw-Hill Publication, 3rd Edition.
4. Herbert Schildt , (2011),Java The Complete Reference, 7th Edition, Publisher McGraw- Hill Osborne Language English, ISBN: 9780071631778
5. Steven Holzner, JAVA 2 Programming Black Book, Wiley India.

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓	✓	✓	✓

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

ELECTIVE COURSE

MCA-DSE-518 (A): Lab on Basics of Web Development

Course Title: Lab on Web Development	Course Type: DSE
Course Code: MCA-DSE-518 (A)	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. Recognize HTML, CSS, and JavaScript syntax and concepts.
2. Explain the purpose and effects of different HTML elements, CSS properties, and JavaScript features.
3. Construct functional webpages and style them with CSS; write JavaScript to handle user interactions and dynamic content.
4. Design and build comprehensive web solutions using HTML, CSS, JavaScript, and SASS, integrating advanced features and ensuring responsiveness and interactivity.

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment
x	✓	✓	x	x	✓

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Create well-structured HTML documents using <div>, <span>, and semantic HTML5 elements like <header>, <nav>, and <section> for accessible layouts. They will build functional forms and embed multimedia content with proper fall-back support.
CO2	Gain proficiency in CSS for styling webpages, utilizing properties, responsive units (e.g., px, em, %, vh), Flexbox for layouts, and animations. They will create theme able designs with CSS variables and compile modular CSS using SASS.
CO3	Handle JavaScript functions, data types, and DOM manipulation, developing interactive applications such as calculators, quizzes, and dynamically updating content based on user input with event listeners.
CO4	Design responsive, accessible web pages for various devices, applying both CSS units and JavaScript interactions to ensure a flexible, user-friendly experience.
CO5	Integrate HTML, CSS, and JavaScript to build interactive, functional web pages, and use browser tools to debug and handle errors effectively across different browsers.

Sr. No.	Assignments
01	Create a Basic HTML Document: 1. Create a simple webpage using correct HTML syntax and document structure, including <!DOCTYPE html>, <html>, <head>, <title>, and <body> tags. 2. Add placeholder text in the body section and display a title in the browser's title bar using the <title> tag
02	Content Structuring: 1. Use <div> tags to structure the webpage into sections (e.g., a header, content area, and footer). 2. Use <span> tags to highlight or style inline elements (e.g., a bolded or colored word in a paragraph).
03	Create a webpage using Semantic HTML5 elements: 1. Design a webpage layout using the following semantic elements: a. <header> for the page header, including a logo and a navigation menu. b. <nav> for a navigation bar with links. c. <section> for the main content sections. d. <article> for individual pieces of content (e.g., a blog post or news item). e. <aside> for sidebar content (e.g., advertisements or related links). f. <footer> for the page footer with contact information and social media links.
04	Create a user registration form: 1. Build a form using the following elements: a. <form> tag to define the form. b. <input> elements for fields like name (text), email (email), and password (password). c. <textarea> for a description or bio input. d. <select> element with multiple <option> choices (e.g., country or role selection). e. <button> to submit the form. 2. Create a multimedia webpage: a. Embed an audio player using the <audio> tag with controls, playing a provided audio file. b. Embed a video player using the <video> tag with controls, autoplay, and loop, using a provided video file. c. Ensure both multimedia elements have fallback content for browsers that do not support them.
05	Style a webpage with various CSS properties: 1. Set the text color of headings (<h1>, <h2>, etc.) using the color property. 2. Apply a background color or image to the body or a specific section using the background property. 3. Change the font family, size, and weight of paragraphs (<p>) using the font-family, font-size, and font-weight properties. 4. Align text in different sections using the text-align property (e.g., center, left, right).
06	Create a webpage to demonstrate typography styling: 1. Use different font families (e.g., serif, sans-serif, monospace) on headings and paragraphs. 2. Set different font sizes for headings (<h1>, <h2>) and body text using font-size.

	3. Apply different font weights (e.g., bold, normal, lighter) to various text elements. 4. Use the font-style property to italicize or emphasize specific text (e.g., using italic for quotes or emphasis).
07	Experiment with absolute and relative CSS units: 1. Create a webpage with sections that use px, em, rem, %, vh, and vw units to define font sizes, widths, and heights. 2. Compare how absolute units (px) and relative units (em, rem) behave when resizing the browser window or changing the font size of the parent element. 3. Set the width of containers using percentage values and adjust the viewport height (vh) and viewport width (vw) to control the size of different sections.
08	Build a photo gallery using Flexbox. 1. The gallery should have the following features: a. The images should automatically wrap to fit the screen size (use flex-wrap). b. Each image should maintain an equal width and height ratio, adapting responsively across various screen sizes. **c.** Use justify-content and align-items properties to center the images when there is extra space.
09	Create an interactive button that changes color, size, and shadow on hover using CSS transitions. 1. Add smooth animations for these properties with a duration of 0.5s and easing effects like ease-in-out for smoother transitions.
10	Create a simple form with input fields (text, email, checkbox, radio buttons, etc.). 1. Style different form elements using attribute selectors (e.g., style input[type="text"] differently from input[type="email"]).
11	Extend the previous assignment to include CSS variables for spacing (e.g., --padding, --margin) and typography (e.g., --font-size, --line-height). 1. Dynamically adjust the layout and typography by modifying the variables inside media queries for responsiveness. 2. Create a simple theme switcher that toggles between two themes (dark and light) by changing the CSS variable values.
12	Install SASS and compile a basic SASS file into CSS. 1. Create a styles.scss file with basic styles and compile it into styles.css. 2. Use SASS features like variables, nesting, and partials to structure your CSS in a more modular way.
13	Create a basic webpage that displays a welcome message and uses JavaScript to: 1. Declare variables of different data types (string, number, boolean, array, object). 2. Display the variables' values on the webpage. 3. Write a function that uses typeof to log the data types of these variables in the browser console.
14	Extend the webpage to include a simple calculator that performs basic arithmetic operations (+, -, *, /): 1. Use operators to perform calculations based on user input. 2. Implement conditional statements to display error messages for invalid inputs (e.g., division by zero). 3. Use loops to display a multiplication table for a number entered by the user.

15	Build a simple webpage where users can enter their name and select their favorite color from a dropdown. 1. Write a JavaScript function to capture user input and display a personalized greeting (e.g., "Hello, [name]!"). 2. Add an event listener to the color dropdown that changes the background color of the page based on the user's selection. 3. Use addEventListener to handle click and change events.
16	Create a basic quiz application with multiple-choice questions: 1. Use conditional statements to validate the answers and give feedback to the user (e.g., "Correct!" or "Try again"). 2. Use loops to display the quiz questions and iterate through an array of answers. 3. Use functions to evaluate and display the user's score at the end of the quiz.
17	Create a webpage with a list of items (e.g., to-do list or shopping list). 1. Use JavaScript to select specific elements from the DOM (e.g., getElementById, querySelectorAll). 2. Modify the list items dynamically using JavaScript (e.g., mark items as completed, change text content, or style). 3. Implement a button that changes the color of all the list items using classList or inline styles.

Tools: *Visual Studio Code, Browser Developer Tools, GitHub, and CodePen.*

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓	✓	✓	✓
End Semester Examination (30)	✓	✓	✓			

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: I

ELECTIVE COURSE

CA-DSE-518 (B): Lab on Java Programming

Course Title: Lab on Java Programming	Course Type: DSE
Course Code: MCA-DSE-518 (B)	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

5. To learn basic Object Oriented Programming concepts of Java Programming Language
6. Programming using inner classes and inheritance, polymorphism and interfaces
7. Develop Modules using Interfaces, Lambda Expressions, Collection, Streams and Files

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Write simple Java Programs with class definitions, basic Java structures and lambda expressions
CO2	Represent the class hierarchy using inheritance, function overloading, function over-riding concepts for assigned problem
CO3	Apply Abstract Classes, Interfaces and packages for developing Functional Models using
CO4	Organise different Data Structures using generic programming
CO5	Implement Different Collection Classes for some datasets
CO6	Develop a project using all Core Java Concepts for given problem

Sr. No.	Assignments
1	**Write and execute Simple Java Programs** **a)** Check if the number is a prime or not **b)** Check if given string is the substring of other string **c)** Occurrence of a word in given text
2	**Write and execute Simple Java Programs using class definition and loops.** a) Read an array on n numbers and print the largest number. b) Read an array of n strings and sort them in ascending order c) Read a list of student (Roll no, name, city, blood group) And print the student data staying in "Jalgaon" (or using any other condition)
3	**Develop programs Using Abstract classes, Interfaces and Lambda Expressions** (Three programs) a) Program that demonstrate using Abstract Class – Shape, that is extended by two different Classes – MyCircle and MyRectangle b) Program that uses Interface for the Vehicle with some **final** attributes and abstract functions. Demonstrate its implementation in Vehicles like – Maruti, Toyota Classes. c) Demonstrate using Lambda expressions for at least 3 different functions. Modify the programs of Experiment a and Experiment b using Lambda Expressions
4	**Develop programs using Inheritance, Cosmic Super class (Object class)** a) Person-Employee/ Person-Student b) Drawing Shapes
5	**Implementation Exceptions, Assertions, and Logging** a) Program using single Exception b) Program using multiple exception c) Program using user defines Exception by extending Exception class d) Program using **Assertions, and Logging for assigned problem**
6	**Implementation Generic Programming:** a) Demonstrate using Generic Programming by defining Stack and its operations b) Demonstrate using Generic Programming by defining Queue and its operations
7	**Demonstrate Collection Classes and Streams** **a)** Programs using Sets, Maps, and Other Collection classes **b)** Use Iterator for the exploring data using collection classes **c)** Demonstrate using multithreading for animation/games
8	**a. demonstrate using graphics ,b. gui component using swing and awt**

Tools: NetBeans or Eclipse platforms of Core Java Programming Java 8 and above

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)			✓	✓	✓	✓
End Semester Examination (30)	✓	✓	✓	✓	✓	✓

Semester-II

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

MCA-DSC-521: Computer Networks

Course Title: Computer Networks	Course Type: DSC
Course Code: MCA-DSC-521	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course covers the major concepts of Computer that mainly includes Core Java concepts with Object Oriented Programming Concepts.

Course Objectives:

1. Understand Fundamental concepts of Computer Communication Networks
2. Define different protocols and related concepts
3. Developing simple applications on networking concepts.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presenta-tion	Guest Ses-sion	Assign-ment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Define different terminology and concepts of Computer Networks
CO2	Understand the Layered architecture, functions of the OSI reference model
CO3	Apply the concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks.
CO4	Analyze various algorithms of different protocols of the TCP/IP Model
CO5	Evaluate datalink layer protocols and routing algorithms regarding their implementation and efficiency
CO6	Develop codes using appropriate language for the algorithms using appropriate data structures

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	2	2	2	1	1	1	1	2	2
CO2	2	2	3	2	2	2	2	3	3	3	2	2
CO3	2	1	1	2	2	2	1	1	1	2	2	2
CO4	3	2	3	3	3	3	2	3	3	3	2	2
CO5	1	1	1	1	2	1	1	1	1	2	2	2
CO6	2	2	2	3	3	2	2	2	3	3	2	2

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Introduction Concepts**: Representation of data and its flow Networks, Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design, Physical Layer Transmission Media, **Analog Transmission**: Modulation Digital data, telephone modem, Modulation of Analog signals. **Digital Transmission**: Line coding scheme, switching methods (circuit switching, Packet switching), Multiplexing: FDM, WDM, TDM	15	5	**CO1** **CO2**
2	**Medium Access sub layer**: Medium Access Sublayer - Channel Allocations, LAN protocols - ALOHA protocols, CSMA, CSMA/CD, Overview of IEEE standards	10	10	**CO2**
3	**Data Link Layer** - Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window	15	10	**CO3**
4	**Network Layer**: Network Layer -IP addressing – IPV4, IPV6, subnet, CIDR, Internetworking, Address mapping – ARP, RARP, BOOTP DHCP–Delivery, Forwarding and Unicast Routing protocols.	10	5	**CO4**
5	**Transport Layer**: - Design issues, connection management, Flow control, TCP window management, congestion control-slow start algorithm, TCP, UDP	5	10	**CO5**
6	**Application Layer**: Data compression, Data Encryption, File Transfer, DNS, HTTP, SMTP, TELNET.	5	10	**CO6**

Reference books:

- Forouzan B. A. (2004), **"Data Communication and Networking",** 4th Edition, McGrawHill.
- Kurose, J.F. and Ross K.W. (2005), **"Computer Networking: A Top-Down Approach Featuring the Internet",** 3rd Edition, Addison-Wesley.
- A.S.Tanenbaum (2006), **"Computer Networks",** 2nd Edition, Prentice Hall India.

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examinatio (60)	✓	✓	✓	✓		

SEMESTER: II

MCA-DSC-522: Operating System Concepts

Course Title: Operating System Concepts	Course Type: DSC
Course Code: MCA-DSC-522	Total Credits: 04
Lectures: Practical: : 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course provides a foundational understanding of operating systems, covering their definition, types, and functions. Students will explore system structures, process management, CPU scheduling, memory management, paging and segmentation, virtual memory, and file systems. Additionally, the course offers an introduction to Linux, including its history, architecture, file system, basic commands, shell scripting, process and user management, GNU C compiler, GNU Profiler, gdb debugger, etc.

Course Objectives:

The subject aims to provide the student with an understanding of basic concepts of operating systems and their architecture, components, etc. It also covers the Linux Operating System and its benefits. Students also learn the Linux commands to implement shell programming applications so that they can develop their own applications in Linux. It includes Linux Binutils, GNU C compiler and debugger.

Teaching/ Evaluation Pedagogy

Chalk & Talk	ICT Tools	Group Discussion	Case Study	Guest Session	Survey	Assignment	Lab
✓	✓	✓	--	✓	--	✓	✓

Course Outcomes: At the end of the Course, the Student will be able to:

CO1	Understand Operating systems' fundamental concepts, functions, and structures.
CO2	Manage processes, process scheduling, CPU scheduling, and deadlock handling.
CO3	Grasp memory hierarchy, allocation techniques, paging, segmentation and virtual memory concepts.
CO4	Study file concepts, access methods, directory disk structures, File sharing and mounting.
CO5	Learn the history, features, and architecture of Linux, perform basic file operations, and write simple shell scripts and Linux Binutils, GNU C Compiler.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	3	3	2	2		2	2				
CO2	2	2	1	1	1		3	3				
CO3	3	2	2	3	3		2	2				
CO4	2	1	2	3	2		1	2				
CO5	2	3	1	2	2		1	1				

UNIT	*Contents of Module*	*Hrs*	*Marks*	*COs*
1	**Unit-1 Introduction to Operating Systems**: Definition, types, and functions of an operating system; System Structures: Operating system services, system calls, system programs, and system structure	6	12	**CO1**
2	**Unit-2 Process Management**: Process concept, process scheduling, operations on processes, inter-process communication; **CPU Scheduling:** Scheduling criteria, scheduling algorithms (FCFS, SJF, Priority, Round Robin, Multilevel Queue Scheduling).	8	15	**CO2**
3	**Unit 3: Deadlocks:** Deadlock characterization, Methods for handling deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	10	12	**CO2**
4	**Unit-4 Memory Management**: Memory Hierarchy, Types of memory, memory allocation techniques; Paging and Segmentation: Basic concepts, paging, segmentation, segmentation with paging; **Virtual Memory:** Demand paging, page replacement algorithms, allocation of frames, thrashing	10	12	**CO3**
5	**Unit-5 File Systems**: File concepts, access methods, directory and disk structure, file system mounting, file sharing, and protection.	10	12	**CO4**
6	**Unit-6 Introduction to Linux**: History, features, and architecture of Linux; **Linux File System:** File and directory structure, file permissions, standard file types; **Basic Commands:** File and directory operations (ls, cp, mv, rm, mkdir), text processing (cat, grep, sort), system status (ps, top, df, du); **Shell Scripting:** Introduction to shell, shell variables, control structures (if, case, while, for), writing simple shell scripts.	10	15	**CO5**
7	**Unit-7 GNU Binutils and C Compiler:** Linux GNU Binary Utilities Binutils commands, GNU Profiler(gprof), GNU C compiler collection and manuals of utilities, using gdb (Debugger)	6	12	**CO5**

Reference Books:

1.Peterson Silberschats, Galvin (2005), Operating System Concepts, Addition Wesley Publication. ISBN-10: 8126554274 ISBN-13: 978-8126554270

2. Peterson, (2007), Linux: Complete Reference, 6th Edition, TMH, ISBN: 9780070222946

3.Foster Johnson Welch, Anderson,(2006),Beginning Shell Scripting, Wiley India (Wrox), ISBN:9780764597916

Assessment Pattern:

Bloom's Category	Remember	Understand	Apply	Analyze	Utilize	Develop
Continuous Internal Evaluation. (40)	✓	✓	✓			✓
End Semester Examination (60)	✓	✓	✓			✓

SEMESTER: II

MCA-DSC-523: Data Structures and Algorithms

Course Title: Data Structures and Algorithms	Course Type: DSC
Course Code: MCA-DSC-523	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course contains core concepts of data structures and algorithms, covering arrays, linked lists, stacks, queues, trees, graphs, and hashing techniques. Students will learn types of data structures, basic data structures along with data representation and common operations performed on the data. Students will learn algorithms for sorting and searching. Emphasis is placed on understanding data representation, management and algorithm design. By the end of the course, students will be equipped with knowledge of various data structures and practical implementations of those structures using any programming language.

Course Objectives:

1. To learn basic concept of data representation and algorithm design and analysis.
2. To learn and understand sequential data structure, searching-sorting and hashing techniques.
3. To learn and understand static and dynamic data representation techniques, linked lists, stack and queue data structures and applications.
4. To learn non-linear data structures Tree and Graph, their algorithms and applications.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Understand the concept of data structure, data representation, algorithm design and analysis.
CO2	Learn and Apply different sorting, searching and hashing algorithms.
CO3	Understand linked lists, stack and queue data structures, difference between static and dynamic data representation, and applications of stack and queue.
CO4	Understand and Implement non-linear data structures trees and graphs, and their applications.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	2	1	2	2	1	1	2	1
CO2	3	3	3	3	2	1	2	2	1	1	2	1
CO3	3	3	3	3	3	1	2	2	1	1	2	1
CO4	3	3	3	3	3	1	2	2	1	1	2	2

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Introduction to Data Structure:** Data, Data Structure Concepts, Types of data structures, Data types, ADT (Abstract Data Type), Algorithm, Algorithm Design Techniques, Algorithm Analysis: Space complexity, Time complexity, Asymptotic Notations (Big O, Omega, Theta)	04	06	1
2	**Arrays:** Array as linear data structure, Representation of array in memory, Operations on Array, List and Strings as ADT, structure and pointer in C/C++.	04	06	1
3	**Sorting, Searching & Hashing: Sorting:** General Background, Bubble Sort, Selection Sort, Insertion Sort. Merge sort. Quick sort and Radix Sort, Comparing time complexities. **Searching:** Linear and Binary search. **Hashing:** Concept, Hashing Techniques, Hash function, Address calculation techniques, Common hashing functions, Collision resolution, Linear Probing, Quadratic, Double hashing. Bucket addressing, Deletion and rehashing.	12	18	2
4	**Linked List:** Introduction, Dynamic representation, Types – Singly, doubly, singly circular, doubly circular, Operations on Linked Lists – Insert, Delete, Traverse, Search, Sort, Reverse, etc.	08	12	3
5	**Stack:** Introduction, Static and Dynamic representation, Operations on stack – PUSH, POP, PEEP, Traverse, Applications of Stack- Infix to Postfix, Evaluation of Postfix expression, Recursion: Definition and Processes, use of stack in recursion. **Queue:** Introduction, Static and Dynamic representation, Operations on queue – Insert, Delete, Traverse, Types of Queues - Circular Queue, Priority Queue and DeQueue.	12	18	3
6	**Tree:** Concept, Tree Data Structure, Tree Terminology, Binary Tree - Representation: Static and Dynamic, Types: Full, Complete, Skewed. Traversal: Recursive and Non-Recursive - Inorder, Preorder, Postorder, Expression Tree: Application – Evaluation of Expression, Heap Tree: Application - Heap Sort, Binary Search Tree: Concept & Operations - Insert, Delete, Traverse. Height Balanced Tree – AVL tree, Concept, Construction, B Tree Concept and Construction.	12	18	4

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
7	**Graph:** Concept, Graph Terminologies, Representation in memory: Adjacency List, Adjacency Matrix, Path Matrix, Weighted Matrix, Traversal: Depth First Search, Breadth First, Search, Spanning Tree, Minimum Spanning Tree Problem-Prim's Algorithm, Shortest Path Problems: Dijkstra's algorithm & Floyd Warshall Algorithm	08	12	4

Reference Books:

1. Horowitz, Sahni, Mehta, (2008), Fundamentals of Data Structures in C++, 2nd Edition, Universities Press, , ISBN 10: 8173716064 ISBN 13: 9788173716065
2. Tenenbaum, Langsam, Augenstein, (1998), Data Structures using 'C', 2nd Edition, Pearson Education, ISBN-10: 8120311779, ISBN-13: 978-0387202778
3. Bala Guruswamy, (2013), Data Structures Using 'C', Tata McGraw Hill Education Private Limited, ISBN-10: 0070701989, ISBN-13: 978-0070701984.
4. Mark A. Weiss, (2002), Data Structures Using 'C', 2nd Edition, Pearson Education India, ISBN-10: 8177583581, ISBN-13: 978-8177583588
5. Seymour Lipschutz, Schaum's Outlines, Data Structures with C, Tata McGraw Hill Education Private Limited, ISBN-10: 938328658X, ISBN-13: 978-9383286584

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	x	x
End Semester Examination (60)	✓	✓	✓	✓	x	x

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

MCA-DSC-524: Lab on Linux Operating System

Course Title: Lab on Linux Operating System	Course Type: DSC
Course Code: MCA-DSC-524	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. To learn basic concepts of Operating systems' fundamental concepts, functions, and structures
2. To learn Linux commands to implement shell programming.
3. To learn and understand Linux Binutils, GNU C compiler and debugger.

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
x	✓	✓	✓	x	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Understand Operating systems' fundamental concepts, functions, and structures.
CO2	Learn basic Linux shell commands.
CO3	Apply Linux commands to perform file and directory operations.
CO4	Implement simple shell scripts and Linux Binutils, GNU C Compiler

Sr. No.	Assignments
1	Use of basic Linux Shell Commands: ls, mkdir, rmdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit.
2	Write a shell script to change the date format. Show the time taken to execute this script
4	Write a shell script to find whether a given number is prime.
5	Write a shell script to count lines, words & characters in its input. (do not use wc)
6	Write a shell script to compute the GCD & LCM of two numbers.

7	Write a shell script to check given number is positive, negative or zero
8	Write a shell script to print the factorial of a given number.
9	Understanding Binutils, its installation, using GNU assembler, using GNU linker, using other binary tools.
10	Show list of files generated while creating an executable using gcc for a C Program(take input a sample c program)
11	Display the object code symbols generated in object code and executables.
12	Show section wise size of an object code and executable using size command.

Tools:

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓			
End Semester Examination (30)	✓	✓	✓			

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

MCA-DSC-525: Lab on Data Structures and Algorithms

Course Title: Lab on Data Structures and Algorithms	Course Type: DSC
Course Code: MCA-DSC-525	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. To learn basic concept of data representation and algorithm design and analysis.
2. To learn and understand sequential data structure, searching-sorting and hashing techniques.
3. To learn and understand static and dynamic data representation techniques, linked lists, stack and queue data structures and applications.
4. To learn non-linear data structures Tree and Graph, their algorithms and applications.

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
x	✓	✓	✓	x	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Understand the concept of data structure, data representation, algorithm design and analysis.
CO2	Apply and use different sorting, searching and hashing algorithms.
CO3	Understand linked lists, stack and queue data structures, difference between static and dynamic data representation, and applications of stack and queue.
CO4	Implement non-linear data structures trees and graphs, and their applications.

Sr. No.	Assignments
1	**Implementation of algorithms based on Arrays:** a) Liner List using Array. Implementing Operations: Insert, Delete, Traverse, Search b) STRING as ADT Implementing Principle operations of a String.
2	**Implementation of algorithms based on Sorting:** a) Bubble sort

	b) Selection sort c) Insertion sort d) Merge Sort e) Quick Sort f) Radix Sort **Implementation of algorithms based on Searching:** a) Linear Search b) Binary Search
3	**Implementation of algorithms based on Linked List:** a) Singly Linked List. Implementing Operations: Insert, Delete, Traverse b) Singly Circular Linked List. Implementing Operations: Insert, Delete, Traverse c) Polynomial arithmetic using linked list. d) Merging of two Linked Lists e) Splitting of Linked List in to two Lists. f) Doubly Linked List. Implementing Operations: Insert, Delete, Traverse
4	**Implementation of algorithms based on Stack:** a) Stacks (Static and Dynamic) b) Application 1: Validation of Arithmetic Expression c) Application 2: Infix to Postfix Conversion of Arithmetic Expression d) Application 3: Evaluation of Postfix Expression e) Application 4: Simulating recursion using stack **Implementation of algorithms based on Queue:** a) Queue (Static and Dynamic) b) Circular Queue (Static and Dynamic) c) Priority Queue (Static and Dynamic) d) DeQueue (Static)
5	**Implementation of algorithms based on Tree:** a) Binary Tree b) Binary Search Tree: Implementation of operations – Search, Insert and Delete c) Binary Tree Traversal Techniques (recursive and non-recursive) i) In-order, ii) Pre-order iii) Post-order d) Heap Tree: Min Heap / Max Heap e) Application – Heap Sort
6	**Implementation of algorithms based on Graph:** a) Depth First Traversal b) Breadth First Traversal c) Obtaining Shortest Path (warshall & Dijkstra's) d) Minimum spanning tree (Prims and Kruskal Algorithms)

Tools: Compilers of any programming language (viz. C / C++ / Java / C# / Python)

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓	✓	✓	X
End Semester Examination (30)	✓	✓	✓	✓	✓	X

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

ELECTIVE COURSE

MCA-DSE-526 (A): Advance Web Development-I

Course Title: Advance Web Development-I	Course Type: DSE
Course Code: MCA-DSE-526 (A)	Total Credits: 04
Lectures: Tutorials: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course delves into advanced web development, covering JavaScript, jQuery, Bootstrap, and Angular. Students will explore object-oriented and asynchronous JavaScript, use jQuery for DOM manipulation and AJAX, and learn to create responsive designs with Bootstrap. The course also introduces the Angular framework for building single-page applications (SPAs). By the end, students will be equipped to develop interactive, responsive web applications.

Course Objectives:

1. Understand advanced JavaScript concepts, including object-oriented programming, asynchronous programming, and error handling techniques.
2. Utilize jQuery for simplified DOM manipulation, animations, and AJAX operations to enhance user interactions in web applications.
3. Implement responsive web design using Bootstrap, understanding its grid system, components, and customization options.
4. Develop modern web applications using Angular, focusing on its environment setup, components, and template-driven architecture.

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment
✓	✓	✓	✓	✓	✓

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	**Apply** object-oriented programming principles, **manage** asynchronous operations using callbacks, promises, and async/await, and **effectively debug** code using error handling techniques and browser developer tools.
CO2	**Write** simplified scripts to **manipulate** the DOM, **create** animations, and **manage** AJAX requests for dynamic content loading, **enhancing** the interactivity and responsiveness of web pages.
CO3	**Develop** responsive and mobile-friendly web pages by **leveraging** Bootstrap's grid system, utilities, and customizable components, including **integrating** Flexbox and **customizing** themes using Sass.

CO4	**Set up** an Angular development environment, **create** and **manage** Angular projects, and **build** modular applications using components, templates, and data-binding for scalable web applications.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	1	3	2	1	1	1	1	1	1
CO2	1	3	2	1	3	1	1	1	1	2	1	1
CO3	3	1	2	3	3	1	1	2	1	2	1	1
CO4	3	3	3	1	3	1	2	2	1	1	2	1

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Unit-1: Advanced JavaScript Concepts** 1.1 Object-Oriented JavaScript. 1.1.1 Objects, Prototypes, and Inheritance. 1.1.2 Constructors and Classes. 1.2 Asynchronous JavaScript: 1.2.1 Introduction to Call-backs, Promises, and Async / Await. 1.2.2 Handling Asynchronous Operations 1.3 Error Handling and Debugging: 1.3.1 Try/Catch Blocks. 1.3.2 Using Browser Developer Tools for Debugging	10	14	CO1
2	**Unit-2: jQuery for Simplified Scripting** 2.1 Introduction to jQuery. 2.1.1 Overview and Setting Up jQuery. 2.1.2 Basic Syntax and Selectors. 2.2 jQuery DOM Manipulation: 2.2.1 Adding, Removing, and Modifying Elements. 2.2.2 Traversing the DOM 2.3 jQuery Effects and Animations: 2.3.1 Hide/Show, Fade, Slide Effects. 2.3.2 Custom Animations with jQuery. 2.4 jQuery AJAX 2.3.3 Introduction to AJAX. 2.3.4 Using jQuery AJAX Methods to Fetch Data.	14	20	CO2
3	**Unit-3: Introduction to Bootstrap** 3.1 Introduction to Bootstrap. 3.1.1 Overview of Bootstrap and its benefits. 3.1.2 Setting up Bootstrap in a web project. 3.2 Bootstrap Grid System 3.2.1 Understanding Rows and Columns. 3.2.2 Responsive Grid Layouts.	10	16	CO3

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
4	**Unit 4: Advance Bootstrap Concepts** 4.1 Flexbox and Grid Layout Integration 4.1.1 Using Bootstrap's built-in Flexbox utilities 4.2 Customizing Bootstrap Components 4.2.1 Using Bootstrap's Sass variables 4.2.2 Creating custom themes 4.3 Bootstrap 5 Flex 4.3.1 Display Flex Utilities, Flex Direction 4.3.2 Flex Wrap, Justify Content	12	18	CO3
5	**Unit-5: Introduction to Angular** 5.1 Overview of Angular Framework 5.1.1 What is Angular, and why use it? 5.1.2 Key features and advantages of Angular. 5.2 Setting Up the Angular Environment 5.2.1 Installing Node.js and Angular CLI. 5.2.2 Creating a new Angular project using Angular CLI. 5.2.3 Understanding the project structure and files. 5.3 Angular Components and Templates 5.3.1 What are components, how do they work in Angular? 5.3.2 Creating a new component using Angular CLI.	14	22	CO4

Reference Books:

1. Eloquent JavaScript: A Modern Introduction to Programming, by Marijn Haverbeke, No Starch Press Publication, 3rd Edition, ISBN-10: 1593279507, ISBN-13: 978-1593279509
2. Pro AngularJS, by Adam Freeman, Apress Publication, 1st Edition, ISBN-10: 1430264489, ISBN-13: 978-1430264484
3. Learning Angular, by Aristeidis Bampakos and Pablo Deeleman, Packt Publishing Publication, 2nd Edition, ISBN-10: 1789955241, ISBN-13: 978-1789955248
4. Online Resources: Mozilla Developer Network (MDN) Web Docs, W3Schools, jQuery Documentation, getbootstrap, AngularJS Documentation.

Tools: *Visual Studio Code, Browser Developer Tools, GitHub, and CodePen.*

Assessment Pattern:

Bloom's Category	Remembe (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓			

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

ELECTIVE COURSE

MCA-DSE-526 (B): Advance Java Programming

Course Title: Advance Java Programming	Course Type: DSE
Course Code: MCA-DSE-526 (B)	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course contains core concepts of familiar with the advanced features of Java Language as generic programming, collection framework.

Course Objectives:

1. To become familiar with the advanced features of Java Language as generic programming, collection framework.
2. To understand RMI technology and concept of reusable components using JavaBeans and EJB.
3. To understand Java Servlets, Java server Pages (JSP) technology and Strut & Hibernate technology
4. Design and develop web based applications using JSP

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Describe the Concepts of components with respect to RMI, JavaBeans and EJB.
CO2	Explain the advance java frameworks like Java Servlets, JSP, hibernate used in web applications
CO3	Apply Strut and Hibernate in the dynamic web applications.
CO4	Analyse the solutions applied with Java Streams and Collection classes
CO5	Evaluate the Java Application based on JDBC and Files
CO6	Develop small Applications using the advance Java Concepts RMI, Javabeans, JSP and Database Connectivity

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	2	2	1	2	2	2	1	1	1
CO2	3	3	3	2	2	2	2	2	2	2	1	1
CO3	1	1	2	2	2	1	2	2	2	1	1	1
CO4	3	3	3	2	2	1	3	3	3	2	3	3
CO5	1	1	1	2	2	3	2	2	2	2	2	2
CO6	2	2	3	3	1	2	2	2	1	3	3	3

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	**Input/Output Streams and Files:** Reading and Writing Bytes, Text Input and Output, Reading and Writing Binary Data, The DataInput and DataOutput Interfaces, Object Input/Output Streams and Serialization. Working with Files-Reading and Writing Files **Database Programming-The** Design of JDBC, The Structured Query Language, JDBC Configuration, JDBC Statements, Prepared Statements, Result sets **Distributed Computing:** Remote Method Invocation- Introduction, Architecture, RMI Object services, stub and Skeleton, steps of developing an RMI system	12	18	**CO1, CO5, CO6**
2	**Java Bean:** Concepts, Writing process, Applications, Properties and Events, Property Editors, Customizer, Persistence; Enterprise JavaBeans: Introduction, Specification, Architecture, Container, Types, Life cycle, Applications	8	18	**CO1, CO2, CO3**
3	**Servlets:** Concepts, Architecture, Servlet Container Writing Process, API, Life Cycle, Hierarchy, ServletConfig, ServletContext, Programming and deployment, Servlet and HTML Form, Session Management, JDBC	10	12	**CO1 CO2**
4	**Java Server Pages:** Introduction, JSP Containers, Architecture, JSP and Servlets. Life Cycle of JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP Environment, JSP Directives, JSP Action, JSP Implicit Objects, JSP Form Processing, JSP Session and Cookies Handling, JSP Session Tracking JSP Database Access, JSP Standard Tag Libraries, JSP Custom Tag, JSP Expression Language, JSP Exception Handling.	12	18	**CO2 CO3**
5	**MVC:** Introduction different types of logic in Java based Web Application, Advantages and Disadvantages of MVC Architecture, **MVC pattern Layer**: Model, View and Controller**. Strut**: Introduction, Understanding Scopes, Custom Tags, The MVC Design Pattern, Simple Validation, Processing Business Logic, Basic Struts Tags, Configuring Struts, validation framework.	8	12	**CO4, CO5**
6	**Overview of Hibernate**: Hibernate Architecture, Hibernate Mapping Types, Hibernate O/R Mapping, Hibernate Annotation, Hibernate Query Language.	10	12	**CO5, CO6**

Reference Books:

1. Cay’s Horstmann and Gary Cornell, (2012), Core Java Volume -1 Fundamentals, 12 th Edition.
2. Horstman Cay, Cornell Gary, Core JavaTM2, Vol.1&amp;2, 11 th Edition, Pearson education.
3. E Balguruswamy, (2000), Programming in Java, Tata McGraw-Hill Publication, 3 rd Edition.
4. Herbert Schildt , (2011),Java The Complete Reference, 7th Edition.
5. Steven Holzner, JAVA 2 Programming Black Book, Wiley India.

Assessment Pattern:

Bloom’s Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓	✓		

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

ELECTIVE COURSE

MCA-DSE-526 (C): Machine Learning

Course Title: Machine Learning	Course Type: DSE
Course Code: MCA-DSE-526 (C)	Total Credits: 04
Lectures: Practical: 4:2	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course provides an introduction to machine learning, covering essential concepts and techniques. Students will gain a thorough understanding of both supervised and unsupervised learning methods, including regression, classification, and clustering. The course also explores probability-based learning techniques and graphical models, equipping students with the skills to analyse and implement a range of machine learning algorithms.

Course Objectives:

1. To introduce students to the basic concepts and techniques of Machine Learning.
2. To have a thorough understanding of the Supervised and Unsupervised learning techniques
3. To understand regression, classification and clustering
4. To study the various probability based learning techniques
5. To understand graphical models of machine learning algorithms

Teaching/ Evaluation Pedagogy

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
✓	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Distinguish between, supervised, unsupervised and semi-supervised learning.
CO2	Apply the apt machine learning strategy for any given problem.
CO3	Suggest supervised, unsupervised or semi-supervised learning algorithms for any given problem.
CO4	Design systems that uses the appropriate graph models of machine learning.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	2	2	2	1	2	1	1	-
CO2	3	3	3	3	2	3	-	1	1	1	-	-
CO3	3	2	3	2	2	3	2	2	-	2	-	1
CO4	3	3	2	3	2	2	2	1	2	-	1	2
CO5	3	3	3	3	2	2	2	1	2	1	-	-
CO6	2	3	3	2	2	2	2	2	1	-	1	1

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
1	Introduction – Programmatic Solution Vs Machine Learning Solution, Components of a Learning Problem, Applications of Machine Learning, designing a Learner, choosing a Model Representation, Inductive Learning, Features, Feature Vector, Feature Space, Instance Space, Hypothesis, Hypothesis Space, Inductive Bias, preference bias and restriction bias, Bias variance, overfitting, under fitting, Bias variance tradeoff, important issues in machine learning, broad types of ML: Supervised, Unsupervised, Reinforcement Learning, Model Parameters and Hyper Parameters. Performance evaluation: confusion matrix (accuracy, precision, recall), cross validation: LOOCV, K-Fold, Stratified K-Fold, time series	16	22	CO1
2	Linear regression – Types of Regression, Residual Error, Estimating Regression Parameters, LMS Update Rule / Widrow-Hoff Learning Rule, Batch Gradient Descent Algorithm, Stochastic Gradient Descent Algorithm. Decision Trees – Entropy, Gain, Gini Index, The ID3 Algorithm, Practical issues in designing DTs, splitting attribute that has continuous data, overfitting, Pre-pruning, post-pruning, reduced error pruning Instance based learning: Voronoi Diagram, Euclidean Distance Manhattan Distance, Minskowaski Distance, kNN, effect of k on kNN performance, Feature reduction: importance of feature reduction, curse of dimensionality, Feature reduction: Selection (filter/wrapper), Extraction (PCA/LDA) Recommender systems – types of recommender systems: content based, collaborative (user based & item-based CF)	16	22	CO1
3	Probability and Bayes learning – Bayes rule, MAP hypothesis, bayes optimal classifier, Gibb’s sampling, NB algorithm (discrete and continuous), Bayesian Networks: Representation, Conditional independence, Inference in Bayesian networks, Learning Bayesian Belief networks.	8	12	CO2

Unit	*Course Contents*	*Hrs*	*Marks*	*COs*
4	Logistic Regression: logistic regression using stochastic gradient ascent, Support Vector Machine: Functional margin, Geometric Margin, Linear SVM: Hard Margin, Soft Margin, Non Linear SVM and Kernel Trick, The SMO (Sequential Minimal Optimization) Algorithm Kernel function and Kernel SVM.	8	14	CO3
5	Clustering: Introduction to clustering, aspects of clustering, major clustering approaches: partitioning, hierarchical, model based, density based, graph theoretic, Quality of Clustering, k-Means clustering algorithm, Time Complexity, advantages & disadvantages, Model based Clustering, introduction to hierarchical clustering, dendrograms, hierarchical agglomerative clustering, single linkage, complete linkage	6	10	CO3
6	Neural network: Perceptron, perceptron training rule, multilayer network, backpropagation, introduction to deep neural network, vanishing/exploding gradient problem and solution, autoencoders, stacked Autoencoders: Introduction, Working, Key functions, Stacked autoencoders (key features, training process, applications, advantages, limitations) CNN: Convolution, kernel, pooling (max/avg), stride, feature map, padding, flattening, RNN: Key Features, Applications, Variants of RNN.	6	10	CO4

Reference Books:

1. Tom Mitchell (1997). Machine Learning. First Edition, McGraw- Hill.
2. Ethem Alpaydin (2009). Introduction to Machine Learning Edition 2. The MIT Press.

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	x
End Semester Examination (60)	✓	✓	✓	✓	✓	x

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

ELECTIVE COURSE

MCA-DSE-527 (A): Lab on Advance Web Development-I

Course Title: Lab on Advance Web Development-I	Course Type: DSE
Course Code: MCA-DSE-527 (A)	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. Students will create and demonstrate JavaScript objects using prototypal inheritance, showing an understanding of object-oriented programming.
2. Students will set up a jQuery project, manipulate the DOM, and implement AJAX to fetch dynamic data.
3. Students will design responsive layouts using Bootstrap's grid system, Flexbox utilities, and customize themes with Sass.
4. Students will develop responsive layouts using Bootstrap's Flexbox and grid utilities, optimizing for different screen sizes.
5. Students will set up Angular development environments, create projects, and manage components for scalable applications.

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment
x	✓	✓	x	x	✓

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	**Apply Object-Oriented Programming in JavaScript**: Students will apply prototypal inheritance to create and extend objects, demonstrating their comprehension of object-oriented programming concepts by building and manipulating JavaScript objects.
CO2	**Implement jQuery for DOM Manipulation and AJAX**: Students will analyze and utilize jQuery to manipulate the DOM, traverse elements, and handle events. They will synthesize AJAX requests to dynamically load and display content, enhancing webpage interactivity.
CO3	**Design and Develop Responsive Webpages using Bootstrap**: Students will create responsive, mobile-friendly layouts using Bootstrap's grid system, Flexbox utilities, and components. They will evaluate and apply different layout strategies to ensure adaptability across devices and screen sizes.

CO4	**Customize and Apply Bootstrap Themes with Sass**: Students will design and customize Bootstrap themes by using Sass variables and mixings, demonstrating creativity and problem-solving skills in personalizing web designs.
CO5	**Set Up and Develop Modular Angular Applications**: Students will configure the Angular development environment, create and manage modular applications using Angular components and templates, demonstrating their ability to build scalable web applications aligned with modern web development standards.

Sr. No.	Assignments
1	Create an Object with Prototypal Inheritance: ○ Create an object `Person` with properties like `name` and `age`, and a method `greet()` that logs a greeting to the console. ○ Create another object Student that inherits from Person using prototypes. Add a method study() to Student that logs a message about studying. Demonstrate inheritance by creating instances of Student and calling both greet() and study() methods.
2	Set up a jQuery project: ○ Create a simple HTML file and include jQuery via a CDN link in the `<head>` section. Verify jQuery is set up correctly by writing a simple script that logs "jQuery is working!" to the console when the document is ready.
3	Using jQuery Selectors: ○ Create an HTML page with several elements, including `<div>`, `<p>`, `<h1>`, and elements with specific `id` and `class` attributes. ○ Write jQuery code to select and manipulate these elements: • Select all paragraphs and change their text color. • Select elements with a specific class and hide them. Select an element by ID and add a border around it.
4	DOM Traversal using jQuery: ○ Create a nested list of items and a paragraph. ○ Use jQuery to traverse and manipulate the DOM: • Select the parent element of a list item and change its background color. • Find all sibling elements of a specific element and hide them. • Navigate to child elements and change their text. and DOM Manipulation with jQuery: ○ Create a webpage with a list (`<ul>`) and a button. ○ Write jQuery code to: • Add a new list item to the <ul> when the button is clicked. • Remove the last list item when another button is clicked. Modify the content of a specific list item dynamically (e.g., change the text of the second item).
5	AJAX Basics: ○ Set up a basic HTML page with a button and an empty div. ○ Use jQuery to send an AJAX request to a server or API (for example, using a public API like "JSONPlaceholder").

	When the button is clicked, use AJAX to fetch data from the server and display it in the div.
6	Create a Basic Webpage with Bootstrap: o Set up a new HTML project and include Bootstrap using a CDN link in the `<head>` section. o Create a simple webpage that uses Bootstrap components such as a navigation bar, a jumbotron or hero section, and a footer. Ensure that the webpage is styled using Bootstrap's classes (e.g., container, row, col) and displays correctly.
7	Design a Responsive Layout with Bootstrap: o Create a webpage layout with multiple sections (e.g., header, main content, sidebar, footer) using Bootstrap's grid classes. o Implement different column layouts for various screen sizes (e.g., 12 columns on mobile, 6 columns on tablets, and 4 columns on desktops). Use Bootstrap's responsive grid classes (e.g., `col-sm-`, `col-md-`, `col-lg-`, `col-xl-`) to ensure the layout is fluid and adapts to different screen sizes.
8	Create a Flexbox Layout Using Bootstrap: o Set up a simple webpage that uses Bootstrap's Flexbox utilities to create a responsive layout. o Design a layout with a header, main content area, and footer. Use Bootstrap's Flexbox classes to align the header and footer and make the main content area flexible. Implement various Bootstrap Flexbox utilities such as `d-flex`, `justify-content-center`, `align-items-start`, and `flex-column` to control the layout and alignment of elements.
9	Develop a Custom Bootstrap Theme: o Build a custom theme by creating a new set of Sass variables and mixins for Bootstrap components (e.g., buttons, cards, alerts). o Design a simple webpage using Bootstrap components that showcase your custom theme. This could include a customized navbar, buttons with new colors, and themed card components. Demonstrate how your theme looks by including various Bootstrap components styled according to your custom theme.
10	Implement Flexbox Layout with Bootstrap 5: o Create a layout that demonstrates the use of Bootstrap 5 Flex utilities. Design a container with multiple items and use `d-flex` to make the container a flex container. Experiment with `flex-direction` classes (`flex-row`, `flex-column`) to change the direction of the flex items.
11	Environment Setup Documentation: o Create a step-by-step guide for installing Node.js and Angular CLI on different operating systems (Windows, macOS, Linux). o Include screenshots or commands used during the installation process. o Verify the installation by checking the versions of Node.js and Angular CLI installed.
12	Create a New Angular Project: o Use Angular CLI to create a new Angular project. o Follow the CLI prompts to set up the project (e.g., project name, routing, stylesheet format). Document the commands used and provide screenshots of the project setup process.

13	Component Overview Document: Write a document explaining what Angular components are and their role in Angular applications.

Tools: *Visual Studio Code, Browser Developer Tools, GitHub, and CodePen.*

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓	✓	✓	✓
End Semester Examination (30)	✓	✓	✓			

SEMESTER: II

ELECTIVE COURSE

MCA-DSE-527 (B): Lab on Web Programming using Java

Course Title: Lab on Web Programming using Java	Course Type: DSE
Course Code: MCA-DSE-527 (B)	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. Study Step-by-Step procedure for building the project in java from ground up by using IDE.
2. Develop application using collection framework, RMI technology, JavaBeans and EJB
3. Develop Web Applications using advanced Java technology Servlets, JSP, Strut and Hibernate

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
x	✓	✓	✓	x	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Step-by-Step procedure for building the project from ground up by using IDE.
CO2	Create dynamic web application to utilize the JavaBeans and EJBs reusable components
CO3	Create web application using servlets, JSP, Strut and Hibernate technologies.

Sr. No.	Assignments
1.	Write a Java program(s) that demonstrates the use of Collection Classes.
2.	Write a Java program that connects to a database using JDBC and does add, delete, modify an retrieve operations. Create Appropriate GUI using awt for user interaction.
2	Write java program(s) that demonstrates generic programming.
3	Write a Java program(s) that demonstrates the use of Collection Classes (Collection frame work).
4	Write a Java program(s) that demonstrates the use of RMI technology.

5	Write a Java program(s) that demonstrates Java Bean.
6	Write a Java program(s) that demonstrates EJB.
7	Write a Java program(s) that demonstrates use of Servlets.
8	Write a Java program(s) that demonstrates use of JSP technology.
9	Implement the dynamic web application(s) to demonstrate use of struts.
10	Implement the dynamic web application(s) to demonstrate use of Hibernate.

Tools: Compilers of any java programming language (NetBeans/Eclipse/ J Developer)

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓	✓	✓	✓
End Semester Examination (30)	✓	✓	✓	✓		

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

ELECTIVE COURSE

MCA-DSE-527 (C): Lab on Machine Learning

Course Title: Lab on Machine Learning	Course Type: DSE
Course Code: MCA-DSE-527 (C)	Total Credits: 02
Lectures: Practical: 4:2	CIE Marks: 20
Lab Hours: 30 Hours	ESE Marks: 30

Course Objectives:

1. Make use of Data sets in implementing the machine learning algorithms
2. Implement various ML algorithms for Classification clustering, regression using a programming language of your choice preferably Python, R-Programming etc.
3. Implement the machine learning concepts and algorithms in any suitable language of choice.

Teaching/ Evaluation Pedagogy:

Class Room Board	ICT Tools	Practical Demo	Presentation	Guest Session	Assignment		
x	✓	✓	✓	✓	✓		

Course Outcomes: At the end of the Course, the Student will be able to –

CO1	Understand the implementation procedures for the machine learning algorithms.
CO2	Design Java/Python programs for various Learning algorithms.
CO3	Apply appropriate data sets to the Machine Learning algorithms.
CO4	Identify and apply Machine Learning algorithms to solve real world problems.

Sr. No.	Assignments
1	Perform Data Pre-processing, Feature Engineering and Exploratory Data Analysis using Python: a) Data Pre-processing: Import Python Libraries (Numpy, Pandas, Matplotlib, Seaborn), Reading Dataset, Data Cleaning/Wrangling, Missing values b) Feature Engineering: Data Reduction, Feature Engineering, Creating Features, Encoding and one-hot-encoding, Feature Scaling: Normalization (MinMaxScaler) and Standardization (StandardScaler), Binning

	c) EDA Exploratory Data Analysis: Statistics Summary, Multivariate Analysis, EDA Univariate Analysis, Data Transformation, EDA Bivariate Analysis, EDA Multivariate Analysis, Impute Missing values, Outlier detection and removal
2	Implement dimensionality reduction using: a) PCA algorithm b) LDA algorithm
3	Implement a program for computing the Bias, Variance and Cross-validation.
4	Implementation of various evaluation metrics using sklearn: Accuracy, Precision, Recall and Confusion Matrix.
5	Implement Simple Linear Regression algorithm using the Gradient Descent Algorithm. (Do not make use of ML libraries like Sklearn)
6	Implement Linear Regression using sklearn library. Use an appropriate data set and calculate the accuracy of your model.
7	Write program to calculate popular attribute selection measures (ASM) like Information Entropy, Information Gain, and Gini Index etc. for decision tree.
8	Using the sklearn library build a decision tree-based classifier (train the classifier using ID3 algorithm). Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
9	Using the sklearn library build a classifier using the k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions. Use the Python ML library classes can be used for this problem.
10	Using the sklearn library build a Naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
11	Using the sklearn library build a logistic regression classifier for the Iris data set stored as a .CSV file. Display the performance of the model in terms of accuracy, precision, recall, F1 Score, AUC and also display the confusion matrix.
12	Write a program to construct a Bayesian network.
13	Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets
14	Implement K-Means Clustering on iris dataset using the scikit-learn (sklearn) library
15	Demonstrates the effect of different metrics on the hierarchical clustering with Agglomerative clustering with different metrics
16	Write a python code for Agglomerative clustering, compute the ward linkage using Euclidean distance, and visualize it using a dendrogram

Tools: Python Compiler 3.8 onward, Jupiter/PyCharm/ any Editor, Libraries NumPy, SciPy, SciKit-learn, Pandas, etc.

Assessment Pattern:

Bloom's Category	Remember (✓)	Understand (✓)	Apply (✓)	Analyze (✓)	Evaluate (✓)	Create (✓)
Continuous Internal Evaluation. (20)	✓	✓	✓	✓	✓	✓
End Semester Examination (30)	✓	✓	✓	✓	✓	✓

KCES's Institute of Management and Research (Autonomous), Jalgaon

Faculty of Science and Technology, School of Computer Applications
M.C.A. (Master of Computer Application) Programme

SEMESTER: II

MCA-RM-528: Research Methodology

Course Title: Research Methodology	Course Type: RM
Course Code: MCA-RM-528	Total Credits: 04
Lectures: Practical:: 4:0	CIE Marks: 40
Lecture Hours: 60 Hours	ESE Marks: 60

Course Description:

This course offers an overview of research methodology including basic concepts employed in quantitative and qualitative research methods. Includes computer applications for research project development.

Course Objectives:

To give an overview of the research methodology and explain the technique of defining a research problem
- To explain the functions of the literature review in research.
- To explain carrying out a literature search, its review, developing theoretical and conceptual frameworks, and writing a review.
- To explain various research designs and their characteristics.
- To explain the details of sampling designs, and also different methods of data collections.
- To explain the art of interpretation and the art of writing research reports.

Teaching/ Evaluation Pedagogy

Chalk & Talk	ICT Tools	Group Discussion	Case Study	Guest Session	Survey	Assignment	Lab
✓	✓	✓	✓	✓	✓	✓	---

Course Outcomes: At the end of the Course, the Student will be able to:

CO1	Identify a suitable research approach for the intended research topic.
CO2	Define the research problems, purpose, and objectives of the research.
CO3	Conduct a systematic literature review on the research topic.
CO4	Demonstrate scientific methods and processes of conducting quantitative and qualitative research.
CO5	Apply appropriate research methodology for a research topic.
CO6	Develop a research model best suitable for the selected research topic.

Mapping of Course Outcomes to Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	3	2	3	2	2	2	2	3	2	1
CO2	2	1	2	1	2	3	3	3	2	2	3	2
CO3	3	2	1	2	2	2	1	2	2	3	3	1
CO4	2	1	1	2	3	3	2	2	3	2	2	2
CO5	1	2	2	2	3	2	2	3	3	2	1	2
CO6	2	2	3	2	1	2	2	2	2	1	2	2

SN	*Contents of Module*	*Hrs*	*Marks*	*COs*
1	**Unit 1: Foundations of Research:** Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable. Research Process.	**8**	**12**	**CO1**
2	**Unit 2: Problem Identification & Formulation:** Research Question, Investigation Question, Literature review methods, Problem identification, Measurement Issues, Hypothesis, Qualities of a Good Hypothesis –Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance.	**8**	**12**	**CO2, CO3**
3	**Unit 3: Research Design:** Concept and Importance in Research, Features of a good research design, Exploratory Research Design, concept, types and uses, Descriptive Research Designs, concept, types and uses. **Experimental Design**: Concept of Independent & Dependent variables.	**8**	**12**	**CO4**
4	**Unit 4: Qualitative and Quantitative Research:** Qualitative research, Quantitative research, measurement concept, causality, generalization, replication. Merging the two approaches.	**8**	**12**	**CO4**
5	**Unit 5: Measurement and Evaluation:** Concept of measurement– what is measured? Problems in measurement in research – Validity and Reliability. Levels of measurement – Nominal, Ordinal, Interval, Ratio.	**8**	**12**	**CO4**
6	**Unit 6: Sampling:** Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non-Response. Characteristics of a good sample. Probability Sample – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage Sampling. Determining the sample size – Practical considerations in sampling and sample size.	**10**	**15**	**CO5**
7	**Unit 7: Data Analysis, Preparation and Research Model Development:** Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis, Cross tabulations, and Chi-square test including testing hypothesis of association, research model development.	**10**	**15**	**CO6**

Reference Books:

1. Business Research Methods – Donald Cooper & Pamela Schindler, TMGH, 9th edition
2. Business Research Methods – Alan Bryman & Emma Bell, Oxford University Press.
3. Research Methodology – C.R.Kothari

Assessment Pattern

Bloom's Category	Remember	Understand	Apply	Analyze	Utilize	Develop
Continuous Internal Evaluation. (40)	✓	✓	✓	✓	✓	✓
End Semester Examination (60)	✓	✓	✓	✓	✓	✓