Pimpri Chinchwad Education Trust's

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR NO. 26, PRADHIKARAN, NIGDI, PUNE 411044

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

Curriculum Structure and Syllabus
Of
First Year Master of Computer Applications (MCA)
(Regulations 2024)

Effective from Academic Year 2025-26 (Version 1.0)

(Updated with minor changes from A.Y. 2025-2026)

Institute Vision

To be one of the top 100 Engineering Institutes of India in coming five years by offering exemplarily Ethical, Sustainable and Value Added Quality Education through a matching ecosystem for building successful careers.

Institute Mission

1. Serving the needs of the society at large through establishment of a state-of-art Engineering Institute.
2. Imparting right Attitude, Skills, Knowledge for self-sustenance through Quality Education.
3. Creating globally competent and Sensible engineers, researchers and entrepreneurs with an ability to think and act independently in demanding situations.

EOMS Policy

“We at PCCOE are committed to offer exemplarily Ethical, Sustainable and Value-Added Quality Education to satisfy the applicable requirements, needs and expectations of the Students and Stakeholders.

We shall strive for technical development of students by creating globally competent and sensible engineers, researchers, and entrepreneurs through Quality Education.

We are committed for Institute’s social responsibilities and managing Intellectual property.

We shall achieve this by establishing and strengthening state-of-the-art Engineering Institute through continual improvement in effective implementation of Educational Organizations Management Systems (EOMS).”

Course Approval Summary

Board of Studies - Department of Master of Computer Applications (MCA)

Sr. No.	Name of Course	Course Code	Page Number	Signature and Stamp of BoS Chairman
1	Software Engineering	MCA31PC01	16	
2	Data Structures	MCA31PC02	18	
3	Data Communication and Networking	MCA31PC03	20	
4	SQL Programming Lab	MCA31PC04	22	
5	Java Programming Lab	MCA31PC05	25	
6	Python Programming Lab	MCA31PC06	36	
7	Research Methodologies and IPR	MCA31EL01	44	
8	Mathematical Foundation for Computer Application-1	MCA31BS01	47	
9	Operating System Concepts	MCA32PC07	50	
10	Web Technologies Lab	MCA32PC08	52	
11	Data Warehouse and Data Mining	MCA32PE01	54	
12	Web Development using Django	MCA32PE02	56	
13	UI/UX Design	MCA32PE03	58	
14	Start-Up and New Venture Development	MCA32PE04	61	
15	Data Science	MCA32PE09	63	
16	Cloud Computing	MCA32PE10	65	
17	Blockchain Technology	MCA32PE11	67	
18	Business Opportunity Identification	MCA32PE12	69	
19	Data Warehouse and Data Mining Lab	MCA32PE05	71	

20	Web Development using Django Lab	MCA32PE06	74	
21	UI/UX Design Lab	MCA32PE07	76	
22	Start-Up and New Venture Development Lab	MCA32PE08	78	
23	Data Science Lab	MCA32PE13	80	
24	Cloud Computing Lab	MCA32PE14	82	
25	Blockchain Technology Lab	MCA32PE15	84	
26	Business Opportunity Identification Lab	MCA32PE16	86	
27	Project Management	MCA32EM01	88	
28	Mathematical Foundation for Computer Application-2	MCA32BS02	90	
29	Research Project	MCA32EL02	92	

Approved by Academic Council:

Chairman, Academic Council

Pimpri Chinchwad College of Engineering

INDEX

CURRICULUM FRAMEWORK

(Regulations 2024)

LIST OF ABBREVIATIONS

Sr. No.	Abbreviation	Type of Course
1	BSC	Basic Science Course
2	PCC	Programme Core Course
3	PEC	Programme Elective Course
4	OEC	Open Elective
5	VSEC	Vocational and Skill Enhancement Course
6	AEC	Ability Enhancement Course
7	EEM	Entrepreneurship/Economics/Management Course
8	ELC	Experiential Learning Courses

COURSE WISE CREDIT DISTRIBUTION

Sr. No.	Type of Course	No. of Courses	Credits	
			No.	%
1	Basic Science Course	2	8	10
2	Programme Core Course	12	24	30
3	Programme Elective Course	7	16	20
4	Open Elective	1	2	2.5
5	Vocational and Skill Enhancement Course	1	2	2.5
6	Ability Enhancement Course	2	4	5
7	Entrepreneurship/Economics/Management Course	2	4	5
8	Experiential Learning Courses	4	20	25
	TOTAL	**31**	**80**	**100**

SEMESTER-WISE COURSE DISTRIBUTION

Course Distribution : Semester Wise						
Sr. No.	**Type of Course**	**No. of Courses / Semester**				**Total**
		1	**2**	**3**	**4**	
1	Basic Science Course	1	1	0	0	2
2	Programme Core Course	6	2	4	0	12
3	Programme Elective Course	0	4	2	1	7
4	Open Elective	0	0	0	1	1
5	Vocational and Skill Enhancement Course	0	0	1	0	1
6	Ability Enhancement Course	0	0	2	0	2
7	Entrepreneurship/Economics/Management Course	0	1	0	1	2
8	Experiential Learning Courses	1	1	1	1	4
	Total	**8**	**9**	**10**	**4**	**31**

SEMESTER-WISE CREDIT DISTRIBUTION

Credit Distribution : Semester Wise						
Sr. No.	**Type of Course**	**No. of Credits / Semester**				**Total**
		1	**2**	**3**	**4**	
1	Basic Science Course	4	4	0	0	8
2	Programme Core Course	12	4	8	0	24
3	Programme Elective Course	0	8	4	4	16
4	Open Elective	0	0	0	2	2
5	Vocational and Skill Enhancement Course	0	0	2	0	2
6	Ability Enhancement Course	0	0	4	0	4
7	Entrepreneurship/Economics/Management Course	0	2	0	2	4
8	Experiential Learning Courses	4	2	2	12	20
	Total	**20**	**20**	**20**	**20**	**80**

Curriculum Structure

First Year

MCA

CURRICULUM STRUCTURE

First Year MCA Semester – I

First Year MCA (Regulations 2024) **(With effect from Academic Year 2025-2026)**																	
Semester-I																	
Course Code	**Course Name**	**Credit Scheme**				**Teaching Scheme (Hours/Week)**					**Evaluation Scheme and Marks**						**Total**
											FA		**SA**	**TW**	**PR**	**OR**	
		L	**P**	**T/A**	**Total**	**L**	**P**	**T/A**	**O***	**Total**	**FA-1**	**FA-2**					
MCA31PC01	Software Engineering	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA31PC02	Data Structures	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA31PC03	Data Communication and Networking	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA31PC04	SQL Programming Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA31PC05	Java Programming Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA31PC06	Python Programming Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA31EL01	Research Methodologies and IPR	3	0	1	4	3	0	1	6	4	20	20	60	0	0	0	100
MCA31BS01	Mathematical Foundation for Computer Application-1	3	0	1	4	3	0	1	6	4	20	20	60	0	0	0	100
Total		**12**	**6**	**2**	**20**	**12**	**12**	**2**	**24**	**26**	**70**	**70**	**210**	**0**	**90**	**60**	**500**

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

O* : Over and Above, Self Study, Experiential Learning

First Year MCA Semester – II

First Year MCA (Regulations 2024) (With effect from Academic Year 2025-2026)																	
Semester-II																	
Course Code	**Course Name**	**Credit Scheme**				**Teaching Scheme (Hours/Week)**					**Evaluation Scheme and Marks**						**Total**
											FA		**SA**	**TW**	**PR**	**OR**	
		L	**P**	**T/A**	**Total**	**L**	**P**	**T/A**	**O***	**Total**	**FA-1**	**FA-2**					
MCA32PC07	Operating System Concepts	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA32PC08	Web Technologies Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA32PE01 to MCA32PE04	Elective-1	2	0	0	2	2	0	0	0	2	10	10	30	0	0	0	50
MCA32PE09 to MCA32PE12	Elective-2	2	0	0	2	2	0	0	0	2	10	10	30	0	0	0	50
MCA32PE05 to MCA32PE08	Elective-1 Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA32PE13 to MCA32PE16	Elective-2 Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA32EM01	Project Management	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA32BS02	Mathematical Foundation for Computer Application-2	3	0	1	4	3	0	1	6	4	20	20	60	0	0	0	100
MCA32EL02	Research Project	0	2	0	2	0	4	0	0	4	0	0	0	50	0	0	50
	Total	**11**	**8**	**1**	**20**	**11**	**16**	**1**	**14**	**28**	**60**	**60**	**180**	**50**	**90**	**60**	**500**

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

O* : Over and Above, Self Study, Experiential Learning

Curriculum Structure

Second Year

MCA

Second Year MCA Semester – I

Second Year MCA (Regulations 2024)
(With effect from Academic Year 2025-2026)

Semester-I

Course Code	**Course Name**	**Credit Scheme**				**Teaching Scheme (Hours/Week)**					**Evaluation Scheme and Marks**						**Total**
											FA						
		L	**P**	**T/A**	**Total**	**L**	**P**	**T/A**	**O***	**Total**	**FA-1**	**FA-2**	**SA**	**TW**	**PR**	**OR**	
MCA33PC09	Advanced Database Management	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA33PC10	Advanced Web Technologies	2	0	0	2	2	0	0	4	2	10	10	30	0	0	0	50
MCA33PC11	Advanced Web Technologies Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA33PC12	Mobile Application Development Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA33PE17 to MCA33PE20	Elective-3	2	0	0	2	2	0	0	0	2	10	10	30	0	0	0	50
MCA33PE21 to MCA33PE24	Elective-3 Lab	0	2	0	2	0	4	0	0	4	0	0	0	0	30	20	50
MCA33AE01	Seminar	0	2	0	2	0	4	0	0	4	0	0	0	50	0	0	50
MCA33AE02	Professional Development Training	0	0	2	2	0	0	2	0	2	0	0	0	50	0	0	50
MCA33VS01 to MCA33VS02	Skill Enhancement	0	2	0	2	0	4	0	0	4	0	0	0	50	0	0	50
MCA33EL03	Mini Project	0	2	0	2	0	4	0	0	4	0	0	0	50	0	0	50
	Total	**6**	**12**	**2**	**20**	**6**	**24**	**2**	**8**	**32**	**30**	**30**	**90**	**200**	**90**	**60**	**500**

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

O* : Over and Above, Self Study, Experiential Learning

Second Year MCA Semester – II

Second Year MCA (Regulations 2024) (With effect from Academic Year 2025-2026)																	
Semester-II																	
Course Code	Course Name	Credit Scheme				Teaching Scheme (Hours/Week)					Evaluation Scheme and Marks						Total
											FA		SA	TW	PR	OR	
		L	P	T/A	Total	L	P	T/A	O*	Total	FA-1	FA-2					
MCA34EM02	Emerging Trends in IT	0	0	2	2	0	0	2	0	2	0	0	0	50	0	0	50
MCA34OE01	Open Elective - MOOC	0	0	2	2	0	0	2	0	2	0	0	0	50	0	0	50
MCA34EL04	Internship	0	12	0	12	0	24	0	0	24	0	0	0	150	0	150	300
MCA34PE25 to MCA34PE28	Elective-4	3	0	1	4	3	0	1	0	4	20	20	60	0	0	0	100
Total		**3**	**12**	**5**	**20**	**3**	**24**	**5**	**0**	**32**	**20**	**20**	**60**	**250**	**0**	**150**	**500**

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

O* : Over and Above, Self Study, Experiential Learning

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

FYMCA (Sem II)			
Course Code	**Elective 1**	**Course Code**	**Elective 2**
MCA32PE01	Data Warehouse and Data Mining	MCA32PE09	Data Science
MCA32PE02	Web Development using Django	MCA32PE10	Cloud Computing
MCA32PE03	UI/UX Design	MCA32PE11	Blockchain Technology
MCA32PE04	Start-Up and New Venture Development	MCA32PE12	Business Opportunity Identification
MCA32PE05	Data Warehouse and Data Mining Lab	MCA32PE13	Data Science Lab
MCA32PE06	Web Development using Django Lab	MCA32PE14	Cloud Computing Lab
MCA32PE07	UI/UX Design Lab	MCA32PE15	Blockchain Technology Lab
MCA32PE08	Start-Up and New Venture Development Lab	MCA32PE16	Business Opportunity Identification Lab

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

SYMCA (Sem I)		SYMCA (Sem II)	
Course Code	**Elective 3**	**Course Code**	**Elective 4**
MCA33PE17	Advanced Data Science	MCA34PE25	Internet of Things
MCA33PE18	Network Security	MCA34PE26	Decision Science
MCA33PE19	Dynamic Application Development and Frameworks	MCA34PE27	Advanced Network Security and Cryptography
MCA33PE20	Startup Management Essentials	MCA34PE28	Entrepreneurial Marketing
MCA33PE21	Advanced Data Science Lab		
MCA33PE22	Network Security Lab		
MCA33PE23	Dynamic Application Development and Frameworks Lab		
MCA33PE24	Startup Management Essentials Lab		

List of Courses – Skill Enhancement Courses (Second Year MCA)

SYMCA (Sem I)			
Course Code	**Course Title**	**Course Code**	**Course Title**
MCA33VS01	Advanced Excel	MCA33VS02	Digital Marketing

Syllabus of Courses
Semester I
First Year MCA

Course Syllabus of Semester – I Courses (w.e.f. Year 2025-26 (Version 1.0))

Program:	MCA							Semester : I	
Course:	Software Engineering							Code : MCA31PC01	
Credits	Teaching Scheme (Hrs. / Week)				Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	O*	FA		TW	SA	Total
					FA-1	FA-2			
2	2	-	-	4	10	10	-	30	50

Prior knowledge of :

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

is essential.

Course Objectives

This course aims at enabling students:

1. To impart concepts of study on the processes and methods of software and agile software development methodology.
2. To learn & understand the Requirement analysis and system Design.
3. To Learn & apply the software design principles to develop software GUI and object-oriented models.
4. To understand the software quality management and testing methodology.

Course Outcomes

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

1. Know the fundamentals of the software development life cycle with current trends in the area of Software Engineering.
2. Apply object-oriented software design and development techniques.
3. Determine an appropriate and effective graphical user interface.
4. Examine the testing design techniques for preparing test cases.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Introduction to Software Engineering** The Linear Sequential Model, Prototyping Model, RAD Model, Spiral Model,Agile Development: Agile manifesto, Agility principles, Extreme Programming, SCRUM, Crystal, Kanban, Feature Driven Development, Adaptive Software Development,Requirement Engineering: Types of Requirements –Functional and Nonfunctional, IEEE standard format Structure and contents of SRS	6
2	**System Design** Unified Modeling Language(UML):Class Diagram and Object, Deployment Diagram, Use case Diagram, Activity Diagram, Sequence diagram, State Transition Diagram,Graphical User Interface: Design patterns. Elements of good design, User Interface Design, Case study on all above diagrams	10

3	**Software Quality Assurance and Testing:** Software Quality Assurance Fundamentals: Quality Assurance, QualityControl, Software Quality Assurance Challenges, Software Quality Metrics: Process Metrics & Product Metrics Software Testing Fundamentals: Introduction of testing, Testing Principles, SoftwareTesting Life cycle, Test plan,Levels of Testing, Test Types,Demo of any one Testing tools: Selenium –WebDriver and Test NG, Appium, JMeter	6
4	**Current trends in Software Engineering** Global software development challenges, Agents and Mobile Software engineering, Artificial Intelligence and Machine Learning in Software Engineering, Low-Code/No-Code Development Platforms, Green software engineering practices.	8
Other*	Agile frameworks like SCRUM and Kanban are implemented using modern tools- Jira,online platforms to build UML diagrams,AI-based testing tool**s**	4
	Total	**30+4=34**

Text Books:

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

Reference Books

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

E-Resources:

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

Program:	**MCA**	**Semester : I**
Course:	**Data Structures**	**Code : MCA31PC02**

Credits	**Teaching Scheme (Hrs. / Week)**				**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**O***	**FA**		**TW**	**SA**	**Total**
					FA-1	**FA-2**			
2	**2**	**-**	**-**	**4**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. Basics of Computer

is essential.

Course Objectives

This course aims at enabling students:

1. To provide the knowledge of basic data structures and their implementations.
2. To understand importance of data structures in context of writing efficient programs.
3. To develop skills to apply appropriate data structures in problem solving

Course Outcomes

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

1. Comprehend fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs
2. Perform basic operations on data structures, including insertion, deletion, traversal, searching, and sorting
3. Apply data structures to solve a variety of computational problems
4. Analyze the time and space complexity of algorithms involving various data structures

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Introduction to Data Structures** Concept, Need, Types, Array Revisited, Time Complexity, Space Complexity	6
2	**Searching and Sorting Techniques** Sorting Algorithms: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, etc, Searching Algorithms : Linear Search, Binary Search, Hashing, etc	8
3	**Linear Data Structures** Stack : Concept, Operations, Applications, Queue : Concept, Operations, Types, Applications, Linked List : Concept, Operations, Types, Applications	8
4	**Non-Linear Data Structures** Tree : Concept, Operations, Types, Applications, Graph : Concept, Representation, Operations, Types, Applications	8
Other*	Self Study : Sorting Techniques, Hashing	4
	Total	**30+4=34**

Text Books:

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

Reference Books:

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

E-Resources:

1. Data Structures in C++ Course, https://www.scaler.com/topics/course/cpp-data-structures/

Program:	**MCA**	**Semester : I**
Course:	**Data Communication and Networking**	**Code : MCA31PC03**

Credits	**Teaching Scheme (Hrs. / Week)**				**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**O***	**FA**		**TW**	**SA**	**Total**
					FA-1	**FA-2**			
2	**2**	**-**	**-**	**4**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

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

is essential.

Course Objectives

This course aims at enabling students:

1. To understand the basic concepts and evolution of data communication and networking.
2. To identify various networking devices and technologies.
3. To analyze error detection and correction methods.
4. To describe network topologies and network communication devices.

Course Outcomes

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

1. Analyze the functioning of data communication and computer network.
2. Evaluate the relevant transmission media and switching techniques.
3. Apply error detection and correction methods to ensure reliable data communication
4. Evaluate the network topologies and network communication devices

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Fundamentals of Data Communication and Networking** Process of data communication and its components: Transmitter, Receiver, Medium, Message, Protocol. Protocols, Standards, Standard organizations, Bandwidth, Data Transmission Rate, Baud Rate and Bits per second. Modes of Communication (Simplex, Half duplex, Full Duplex). Analog Signal and Digital Signal, Analog and Digital Transmission: Analog To Digital , Digital To Analog Conversion Network Architecture: Peer To Peer, Client Server Network	8
2	**Transmission Media and Switching** Communication Media: Guided Transmission Media: Twisted-Pair Cable, Coaxial Cable Fiber - Optic Cable Unguided Media: Radio Waves, Microwaves, Infrared, Satellite Line-of-Sight Transmission Point to Point, Broadcast Multiplexing:Frequency Division Multiplexing, Time Division Multiplexing Switching: Circuit Switched Network, Packet Switched Network	8

3	**Error Detection, Correction and Wireless Communication** Types of Errors: Single Bit Error and Burst Error, Redundancy Error Detection: Longitudinal Redundancy Check (LRC), Vertical Redundancy Check (VRC), Cyclic Redundancy Check(CRC) Forward Error Correction: Forward error Correction IEEE standards: 802.1, 802.2, 802.3, 802.4, 802.5 Wireless LANs: 802.1 1 Architecture, MAC Sublayer, Addressing Mechanism Bluetooth Architecture: Piconet, Scatternet, Mobile Generations: IG, 2G, 3G, 4G and 5G	8
4	**Network Topologies And Network Devices** Network Topologies: Introduction, Definition, Selection, Criteria, Types of Topology- i) Bus ii) Ring iii) Star iv)Mesh v)Tree vi)Hybrid Network Connecting Devices: Hub, Switch, Router, Repeater, Bridge, Gateway, Modem, Wireless infrastructure	6
Other*	Cisco Packet Tracer; Wireshark (Network protocol analyzer; learn how to inspect packet traffic); Understand VPC (Virtual Private Cloud), subnets, and IP addressing in cloud	4
	Total	**30+4=34**

Text Books:

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

Reference Books:

1. C. E. Douglas,Internetworking with TCP/IP Principles, Protocols and Architectures, PHI Learning Pvt Ltd, Delhi,2013

E-Resources:

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

Program:	**MCA**	**Semester : I**
Course:	**SQL Programming Lab**	**Code : MCA31PC04**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Basics of Programming
2. Set Theory

is essential.

Course Objectives

This course aims at enabling students:

1. To develop database handling, data manipulation and data processing skills through SQL & PL/SQL,
2. To develop data centric computer applications.

Course Outcomes

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

1. Analyze the problem of a real-life situation and draw the ER Diagram
2. Create relational database schemas using SQL.
3. Apply SQL queries to retrieve, manipulate and analyze data effectively in relational databases.
4. Implement advanced SQL techniques such as joins and subqueries
5. Create stored procedures and functions for data management tasks

Guidelines: Students will be assessed based on

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

Following Topics will be covered:

- Create ERD and Design Database from ERD
- Introduction to SQL: DDL/DML/TCLstatements, logical,Relational and special Operators,order by clause
- Aggregate functions:MIN, MAX, AVERAGE, SUM, COUNT and Group By…Having Clause.
- Joins, Types of Joins and Nested Subquery
- Introduction to PL/SQL (Block Structure, Control Statement,Exception Handling)
- Introduction to PL/SQL (Creating Procedure & Functions)

Students are advised to use open source/ freeware tool (Oracle Express Edition/ Oracle Live SQL)

Detailed Syllabus

Assign. No.	**Suggested List of Assignments**
1	**ERD & Database Design** Case Study: Real-Life Scenario Analysis for Database Design - Identifying entities, relationships between entities, attributes, keys, cardinalities, generalization, specialization

	etc. Create an Entity-Relationship Diagram illustrating the entities and their relationships in the scenario using ERD Plus, ER Win etc. and create the database tables corresponding to the identified entities and relationships. Ensure that the SQL schema reflects the relationships and constraints identified in the ERD.
2	**Introduction of SQL** Consider the following Schema EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO) DEPT (DEPTNO, DNAME, LOC) Create a database applying suitable constraints and write at least 20 SQL queries using SQL DDL, DML statements and operators
3	**Aggregate functions** Consider the following Schema given below: EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO) DEPT (DEPTNO, DNAME, LOC) Write at least 20 SQL queries to analyze the data by using Aggregate functions-MIN, MAX, AVERAGE, SUM, COUNT and Group By…Having Clause.
4	**Joins, Types of Joins and Nested Subquery** Consider the relational database of the following schema. SAILORS (Sid, Sname, Rating, DOB) RESERVES (Sid, Bid, Date_of_Booking) BOATS (Bid, Boat_Name, Color) Create a database applying suitable constraints and write at least 20 SQL queries to retrieve data from multiple tables using Joins and Nested Subquery.
5	**Introduction to PL/SQL (Structure,Control Statement)** **a)** Write a PL/SQL program to print the multiplication table of a given number k up to 10 using a FOR loop. **b)** Write a PL/SQL program to find the factorial of a given number x using a WHILE loop. **c)** Write a PL/SQL program to calculate the sum of all even numbers from 1 to a given number n. **d)** Write a PL/SQL program to calculate the sum of numbers in a sequence until the cumulative sum exceeds 1000. **e)** Write a PL/SQL program to iterate through the numbers from 1 to a given number n and print each number until you encounter a number that is divisible by 7. Exit the loop once such a number is found. **f)** Write a PL/SQL block to categorize employees into salary grades based on their salaries. If the salary is above [amount], assign "High", if it is between [range], assign "Medium", otherwise assign "Low". Display the employee names along with their assigned salary grades.
6	**Introduction to PL/SQL (Creating Procedure & Functions)** a) Consider the following Schema : EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, RATING) Write a PL/SQL procedure named **calculate_bonus** to calculate the bonus amount for employees based on their performance ratings. The Procedure should accept the employee's salary and performance rating as input parameters. The bonus amount should be determined as follows: ● For performance rating 1, the bonus is 5% of the salary.

- For performance rating 2, the bonus is 7% of the salary.
- For performance rating 3, the bonus is 10% of the salary

Write an exception if EMPNO does not exist in the database or Salary is Null.

b) Consider the following database schema given below
Product (**Product_ID** , Product_Name, Unit_Price)
Promotions (PromotionID , Discount , StartDate, EndDate , **Product_ID**)

Create a database applying suitable constraints and write a PL/SQL function named Calculate_Price to calculate the price of a product. The function should take the product ID and quantity as input parameters. It should retrieve the unit price of the product from the database and calculate the total price based on the quantity ordered. Also, apply any applicable discounts or promotions stored in the database. The function should return the final price of the product.

References:

1. Gruber, M..*Understanding SQL*. BPB,(1990)
2. Bayross, I.*SQL- PL/SQL*. BPB Publications.(2009)
3. Feuerstein, S. *Oracle PL/SQL Best Practices*. O'Reilly Media.(2001)
4. George Koch, Kevin Loney, *Oracle – The Complete Reference*. Oracle Press.(1997)
5. **SQL Tutorial**: https://www.w3schools.com/sql/
6. **SQL Tutorial:** https://www.tutorialspoint.com/sql/index.htm
7. **SQL Tutorial**: https://www.javatpoint.com/sql-tutorial

Program:	**MCA**						**Semester : I**
Course:	**Java Programming Lab**						**Code : MCA31PC05**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Computer Fundamentals
2. Basics Concepts of Programming

is essential.

Course Objectives

This course aims at enabling students:

1. To learn why Java is useful for the design of desktop and web applications.
2. To learn how to implement object-oriented designs and identify Java language components and how they work together in applications.
3. To learn about the concepts and principles of Java programming.
4. To develop applications using object-oriented programming concepts of Java.

Course Outcomes

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

1. Describe different concepts such as Programming Constructs, Multithreading, OOPs, File Handling, Collections using Java Programming.
2. Implement different concepts of object-oriented programming.
3. Implement Annotations, Lambda Expression using Java Programming.
4. Perform different operations related to file handling, multithreading using Java Programming.
5. Use different classes and interfaces from Collection Framework.

Guidelines:

Students will be assessed based on

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

Students are advised to use:

1. Eclipse

Detailed Syllabus

Assign. No.	**Suggested List of Assignments**
1	**Assignments based on Control Structures**. 1. Write a Java program to check whether a given number is even or odd using an if-else statement. 2. Write a Java program that takes a student's score as input and assigns a grade (A, B, C, D, F) using nested if-else statements. 3. Write a Java program to check whether a given year is a leap year or not using a series of if-else conditions.

	4. Write a Java program to find the largest of three numbers using nested if-else statements. 5. Write a Java program that takes two numbers and an operator (+, -, *, /) as input and performs the corresponding operation using a switch statement. 6. Write a Java program to calculate the sum of the first N natural numbers using a for loop. 7. Write a Java program to calculate the sum of the digits of a given number using a while loop. 8. Write a Java program to print the first N numbers in the Fibonacci series using a for loop. 9. Write a Java program to check if a given number is a prime number using a for loop and if-else statements. 10. Write a Java program to display the multiplication table for a given number using a while loop.
2	**Assignments based on Arrays**. 1. Write a Java program to initialize an array with 10 integers and display all the elements. 2. Write a Java program to calculate the sum of all elements in an integer array. 3. Write a Java program to find the largest element in an array of integers. 4. Write a Java program to find the smallest element in an array of integers. 5. Write a Java program to count the number of even and odd numbers in an integer array. 6. Write a Java program to reverse the elements of an array without using a second array. 7. Write a Java program to find all duplicate elements in an integer array. 8. Write a Java program to remove duplicate elements from an integer array and display the new array. 9. Write a Java program to sort an integer array using the bubble sort algorithm. 10. Write a Java program to merge two arrays into a single array.
3	**Assignments based Strings and String Buffers.** 1. Write a Java program to find the length of a given string using the length() method. 2. Write a Java program to count the frequency of each character in a string. 3. Write a Java program to check if a given string is a palindrome using both String and StringBuffer. 4. Write a Java program to concatenate two strings using both the + operator and the concat() method. 5. Write a Java program to reverse a string using a StringBuffer and the reverse() method. 6. Write a Java program to extract a substring from a given string using the substring() method. 7. Write a Java program to replace all occurrences of a specific character in a string with another character using the replace() method. 8. Write a Java program to count the number of vowels and consonants in a string. 9. Write a Java program to split a string into an array of substrings using a delimiter with the split() method. 10. Write a Java program to check if two given strings are anagrams of each other.

4	**Assignments based on Constructors**. 1. Write a Java program to define a class Student with attributes name and age. Create objects of the class and initialize them with values. Print the details of each student. 2. Write a Java program to create a class Car with a default constructor that initializes attributes make and year. Create an object of the class and display the initialized values. 3. Write a Java program to define a class Rectangle with a parameterized constructor to initialize its length and width. Calculate and display the area of the rectangle. 4. Write a Java program to create a class Circle with overloaded constructors—one default and one parameterized. Use the parameterized constructor to set the radius and calculate the area. 5. Write a Java program to create a class Book with a copy constructor to copy the attributes of one book object to another. 6. Write a Java program to create a class Person with attributes name and age. Use the this keyword to differentiate between instance variables and parameters in the constructor. 7. Write a Java program to create a class Employee with a static variable companyName and non-static variables name and salary. Display the details of employees along with the company name. 8. Write a Java program to create a class Product with private attributes id, name, and price. Provide public getter and setter methods to access and update these attributes. 9. Write a Java program to create a class Point with coordinates x and y. Write a method that takes another Point object as a parameter and calculates the distance between the two points. 10. Write a Java program to create a class Student with attributes name and grade. Create an array of Student objects and initialize them with values. Display the details of each student.
5	**Assignments based on Inheritance**. 1. Write a Java program to demonstrate single inheritance by creating a base class Animal with a method eat() and a derived class Dog that inherits from Animal and has its own method bark(). 2. Write a Java program to demonstrate multilevel inheritance by creating a base class Vehicle, a derived class Car that extends Vehicle, and another derived class ElectricCar that extends Car. Include relevant methods in each class and demonstrate calling them. 3. Write a Java program to demonstrate hierarchical inheritance by creating a base class Shape and two derived classes, Circle and Rectangle, each having methods to calculate the area. 4. Write a Java program to demonstrate method overriding by creating a base class Employee with a method getDetails(). Override this method in a derived class Manager to provide additional details. 5. Write a Java program to demonstrate the use of the super keyword to invoke the parent class's constructor and methods in a derived class Student that extends Person.

	6. Write a Java program to demonstrate constructor chaining in inheritance by creating a class A with a constructor and a derived class B with its own constructor. Use super() to chain constructors. 7. Write a Java program to create an abstract class Shape with an abstract method draw(). Create two subclasses, Circle and Square, that provide implementations for the draw() method. 8. Write a Java program to create an interface Printable with a method print(). Create a class Document that implements the Printable interface and provides the implementation for the print() method. 9. Write a Java program to demonstrate multiple inheritance using interfaces by creating two interfaces, Readable and Writable, each with one method. Create a class File that implements both interfaces. 10. Write a Java program to demonstrate a superclass reference holding a subclass object by creating a superclass Animal and subclass Dog. Use an Animal reference to point to a Dog object and call the methods.
6	**Assignments based on Polymorphism, Encapsulations and Abstraction**. 1. Write a Java program to demonstrate method overloading by creating a class MathOperations with overloaded methods add(int, int), add(double, double), and add(int, int, int). 2. Write a Java program to demonstrate method overriding by creating a superclass Vehicle with a method start(). Create a subclass Car that overrides the start() method. 3. Write a Java program to demonstrate runtime polymorphism by creating a superclass Animal with a method makeSound(). Create subclasses Dog and Cat that override makeSound(). Use a superclass reference to call the overridden methods. 4. Write a Java program to create an interface Drivable with a method drive(). Create two classes, Car and Bike, that implement the Drivable interface and provide their own implementation for drive(). 5. Write a Java program to create an abstract class Shape with an abstract method draw(). Create two subclasses, Circle and Rectangle, that provide specific implementations of the draw() method. Demonstrate polymorphism using a reference of type Shape. 6. Write a Java program to create an interface Playable with a method play(). Create classes Football and Basketball that implement the interface. Use polymorphism to call the play() method. 7. Write a Java program to demonstrate upcasting and downcasting with a superclass Fruit and subclasses Apple and Banana. Perform upcasting and then downcasting, and demonstrate any associated risks. 8. Write a Java program to demonstrate dynamic method dispatch by creating a superclass Media with a method play(). Create subclasses Video and Audio that override play(). Use dynamic method dispatch to call these methods.

	9. Write a Java program to create a class Student with private attributes name and age. Provide public getter and setter methods to access and update these attributes. Demonstrate encapsulation by creating an instance and modifying the private attributes through getters and setters. 10. Write a Java program to create a class BankAccount with private attributes accountNumber, balance, and accountHolderName. Use setters to validate input data (e.g., balance cannot be negative) before updating the attributes.
7	**Assignments based on Exception Handling**. 1. Write a Java program that takes two integers as input and divides them. Handle any potential ArithmeticException (such as division by zero) using a try-catch block. 2. Write a Java program to demonstrate multiple catch blocks. Handle both ArrayIndexOutOfBoundsException and NullPointerException in the program. 3. Write a Java program to demonstrate nested try-catch blocks. Use nested blocks to catch ArithmeticException and ArrayIndexOutOfBoundsException. 4. Write a Java program that demonstrates the use of the finally block. Include a finally block to close a Scanner object after reading input from the user, regardless of whether an exception occurs. 5. Write a Java program to demonstrate exception propagation. Create a method that throws an ArithmeticException, and another method that calls this method and handles the exception. 6. Write a Java program to handle multiple exceptions in a single catch block (using Java 7 and above feature). Handle NumberFormatException and ArithmeticException in the same catch block. 7. Write a Java program to demonstrate the use of the throw keyword to explicitly throw an exception. Throw an IllegalArgumentException if the user inputs a negative number. 8. Write a Java program that uses the throws keyword to declare that a method may throw a FileNotFoundException. Read from a file that may not exist and handle the potential exception in the calling method. 9. Write a Java program to demonstrate handling of checked exceptions by reading a file using FileReader. Use try-catch blocks to handle IOException. 10. Write a Java program to demonstrate try-with-resources by reading from a file using BufferedReader. Ensure that the resource is closed automatically, even if an exception occurs.
8	Assignments based on Interface. 1. Write a Java program to create an abstract class Animal with an abstract method makeSound(). Create two subclasses, Dog and Cat, that provide concrete implementations of the makeSound() method. 2. Write a Java program to create an abstract class Vehicle with both abstract and concrete methods. Include an abstract method move() and a concrete method startEngine(). Create subclasses Car and Bicycle to implement the abstract method. 3. Write a Java program to demonstrate that an abstract class can have a

	constructor. Create an abstract class Person with a constructor and a subclass Student that calls the superclass constructor. 4. Write a Java program to create an abstract class Shape with a field color and an abstract method calculateArea(). Create subclasses Circle and Rectangle that provide concrete implementations for the abstract method. 5. Write a Java program to create an abstract class Account with an abstract method calculateInterest(double balance). Create subclasses SavingsAccount and CheckingAccount that implement the method differently. 6. Write a Java program to create an interface Playable with a method play(). Create two classes, Music and Video, that implement the Playable interface and provide concrete implementations of the play() method. 7. Write a Java program to create an interface AnimalActions with methods eat() and sleep(). Create classes Lion and Elephant that implement the interface and provide concrete implementations for both methods. 8. Write a Java program to create two interfaces, Flyable and Swimmable, with methods fly() and swim() respectively. Create a class Duck that implements both interfaces and provides concrete implementations. 9. Write a Java program to create an interface Calculator with a default method add(int a, int b) and an abstract method multiply(int a, int b). Create a class AdvancedCalculator that implements the interface and overrides the abstract method. 10. Write a Java program to create an interface Utils with a static method isNull(Object obj) that checks if an object is null. Demonstrate the usage of the static method in a separate class.
9	Assignments based on Packages. 1. Write a Java program to create a package named shapes and define a class Circle within it. The class should include methods to calculate the area and circumference of a circle. Create another class in a different package that uses the Circle class. 2. Write a Java program to create two packages, company and company.hr. In company, define a class Employee with various fields having different access modifiers (public, protected, default, and private). In company.hr, create a class HRManager that tries to access these fields and demonstrate which fields are accessible. 3. Write a Java program to create a package utilities and define two classes Calculator and Converter in it. Use these classes in a main program to perform various calculations and conversions. 4. Write a Java program to create nested packages org.example.project.models and org.example.project.services. Create a class User in the models package and a class UserService in the services package. Use UserService to perform operations on User. 5. Write a Java program that imports a specific class from a package you created, and another program that imports the entire package using import

	packageName.*. Demonstrate the differences. 6. Write a Java program to demonstrate the use of static import. Create a class MathOperations with static methods add(), subtract(), and multiply(). Use static import to access these methods in another class without using the class name. 7. Write a Java program to demonstrate how to handle naming conflicts when two packages have classes with the same name. Create two packages, each with a class named Utils, and show how to use both classes in the same program. 8. Write a Java program to create a package banking and define an interface BankAccount with methods deposit(), withdraw(), and getBalance(). Implement this interface in two classes SavingsAccount and CheckingAccount. 9. Write a Java program to use an external package like java.util. Demonstrate the usage of classes like ArrayList and HashMap from this package in your program. 10. Write a Java program to create a package ecommerce with classes Product, Customer, and Order. Use Javadoc comments to document the package and its classes. Generate the HTML documentation using the Javadoc tool.
10	Assignments based on File Handling-1 1. Write a Java program to create a text file named output.txt and write some text into it. Use FileWriter and BufferedWriter to perform the write operation. 2. Write a Java program to read the contents of a text file named input.txt and print it to the console. Use FileReader and BufferedReader to perform the read operation. 3. Write a Java program to append text to an existing file named log.txt. Use FileWriter in append mode and BufferedWriter to append additional lines of text. 4. Write a Java program to copy the contents of one file named source.txt to another file named destination.txt. Use FileInputStream and FileOutputStream to perform the copy operation. 5. Write a Java program to check if a file named data.txt exists in the current directory. Print a message indicating whether the file exists or not using the File class. 6. Write a Java program to delete a file named oldfile.txt from the file system. Use the File class to delete the file and print a message indicating success or failure. 7. Write a Java program to rename a file named temp.txt to newfile.txt. Use the File class to perform the renaming operation. 8. Write a Java program to list all the files and directories in a specified directory. Use the File class to get the list of files and directories and print

	their names. 9. Write a Java program to display the size and properties of a file named example.txt. Use the File class to retrieve and display the file size, last modified date, and whether it is a directory or a file. 10. Write a Java program to read a file named data.csv line by line and print each line to the console. Use BufferedReader and handle any potential exceptions.
11	Assignments based on File Handling.-2 1. Write a Java program to serialize an object of a class Person with fields name and age and write it to a file named person.ser. Use ObjectOutputStream for serialization. 2. Write a Java program to deserialize an object from a file named person.ser and print its attributes. Use ObjectInputStream for deserialization. 3. Write a Java program that handles FileNotFoundException and IOException while performing file read and write operations. Include appropriate error messages in the catch blocks. 4. Write a Java program to use java.nio.file package to read the contents of a file named sample.txt using Files.readAllLines(). Print the contents to the console. 5. Write a Java program to create a directory named newdir, and then create a file named file.txt inside this directory. Write some text to the file and read it back. 6. Write a Java program to use the Path and Files classes to copy a file from one location to another. Handle any potential exceptions during the copy operation. 7. Write a Java program to use BufferedInputStream and BufferedOutputStream to read from and write to binary files. Demonstrate how buffered streams can improve performance. 8. Write a Java program to use FileChannel to read data from and write data to a file named data.bin. Use FileChannel's methods for efficient file I/O operations. 9. Write a Java program to read data from a CSV file named contacts.csv and write data to a new CSV file named contacts_copy.csv. Use FileReader, BufferedReader, FileWriter, and BufferedWriter. 10. Write a Java program to monitor changes to a directory using WatchService. Detect and respond to file creation, modification, and deletion events in the specified directory.
12	Assignments based on Multithreading 1. Write a Java program to create a thread by extending the Thread class and overriding its run() method. Start the thread and print a message from within the run() method. 2. Write a Java program to create a thread by implementing the Runnable interface. Implement the run() method and use a Thread object to execute the Runnable. 3. Write a Java program to demonstrate the use of Thread.sleep() to pause a thread for a specific duration. Create two threads where one thread prints

	numbers and the other prints letters, with a delay between each print. 4. Write a Java program to set and display the priority of different threads. Create multiple threads with different priorities and observe their execution order. 5. Write a Java program to demonstrate the use of Thread.join() to ensure that one thread waits for another thread to complete its execution before proceeding. 6. Write a Java program to create and start multiple threads that perform different tasks (e.g., printing numbers, letters, and symbols). Use both the Thread class and the Runnable interface. 7. Write a Java program to display the state of a thread at different stages of its lifecycle (e.g., NEW, RUNNABLE, WAITING, TERMINATED). Use the Thread.getState() method to print the thread states. 8. Write a Java program to create a daemon thread that performs periodic tasks (e.g., printing a message every second). Demonstrate how the daemon thread behaves when the main thread exits. 9. Write a Java program to synchronize access to a shared resource (e.g., a counter) using the synchronized keyword. Create multiple threads that increment the counter and ensure that the increments are performed safely. 10. Write a Java program to demonstrate thread communication using wait() and notify(). Create two threads where one thread produces data and the other thread consumes the data.
13	Assignments based on Collection Framework-1 1. Write a Java program to use the Callable interface and Future class to perform a task that returns a result. Create a Callable task that performs a computation and retrieves the result using Future. 2. Write a Java program to create a thread pool using ExecutorService and submit multiple tasks for execution. Use invokeAll() to wait for all tasks to complete and invokeAny() to get the result of the first completed task. 3. Write a Java program to use concurrent collections such as ConcurrentHashMap and CopyOnWriteArrayList to safely manage shared data between threads. 4. Write a Java program to use the Fork/Join framework to perform a parallel computation. Create a recursive task that divides a large problem into smaller sub-tasks and combines the results. 5. Write a Java program to demonstrate the use of Semaphore to control access to a limited resource. Create a scenario where multiple threads try to access a shared resource with limited permits. 6. Write a Java program to use CountDownLatch to coordinate the completion of multiple threads. Create a scenario where the main thread waits for a specified number of worker threads to finish their tasks. 7. Write a Java program to use CyclicBarrier to synchronize multiple threads

	at a common barrier point. Create a scenario where threads perform a task and wait at the barrier before proceeding. 8. Write a Java program to demonstrate the use of ThreadLocal to maintain thread-specific data. Create a ThreadLocal variable and show how each thread maintains its own instance of the variable. 9. Write a Java program to solve the producer-consumer problem using a shared buffer and synchronization. Implement both producer and consumer threads that interact with the buffer. 10. Write a Java program to simulate and detect a deadlock situation. Create multiple threads that acquire multiple locks and demonstrate how a deadlock can occur. Implement strategies to avoid or resolve deadlocks.
14	Assignments based on Collection Framework-2. 1. Write a Java program to create an ArrayList of String elements. Add, remove, and retrieve elements from the list. Demonstrate various methods such as add(), remove(), get(), and size(). 2. Write a Java program to create a LinkedList of Integer elements. Add and remove elements from the list and demonstrate the difference between ArrayList and LinkedList in terms of performance for these operations. 3. Write a Java program to create a LinkedList of Integer elements. Add and remove elements from the list and demonstrate the difference between ArrayList and LinkedList in terms of performance for these operations. 4. Write a Java program to create a TreeSet of Integer elements and sort the elements in ascending order. Demonstrate how TreeSet maintains a sorted order. 5. Write a Java program to create a HashMap with Integer keys and String values. Add key-value pairs to the map, retrieve values based on keys, and remove entries. 6. Write a Java program to create a TreeMap with String keys and Double values. Display the map in sorted order based on keys. 7. Write a Java program to create a LinkedHashMap and demonstrate how it maintains the insertion order of elements. Add entries to the map and iterate over them. 8. Write a Java program to create a PriorityQueue of Integer elements. Add elements to the queue and demonstrate how the queue orders elements based on their natural ordering. 9. Write a Java program to create an ArrayDeque and use it as both a stack and a queue. Demonstrate the use of methods like push(), pop(), offer(), and poll(). 10. Write a Java program to demonstrate different ways to iterate over a List, Set, and Map. Use enhanced for loops, iterators, and lambda expressions for iteration.

15	Assignments based on Collection Framework-3. 1. Write a Java program to create a custom class Person that implements the Comparable interface. Sort a List<Person> based on a natural ordering of the Person objects. 2. Write a Java program to create a custom class Employee and implement a Comparator to sort a list of employees by their salary. Use Collections.sort() with the custom comparator. 3. Write a Java program to demonstrate the use of the Collections utility class. Use methods like sort(), shuffle(), reverse(), and binarySearch() on various collections. 4. Write a Java program to convert a List to a Set and vice versa. Use HashSet and ArrayList for the conversions and demonstrate how duplicates are handled in the conversion. 5. Write a Java program to create a generic class Box<T> that can hold any type of object. Create instances of Box for different types and demonstrate adding and retrieving objects. 6. Write a Java program to demonstrate the use of thread-safe collections such as ConcurrentHashMap and CopyOnWriteArrayList. Show how these collections can be used safely in a multithreaded environment. 7. Write a Java program to use EnumSet and EnumMap with an enum type. Demonstrate how EnumSet can be used to store a set of enum values and EnumMap for mapping enum keys to values. 8. Write a Java program to implement a custom collection class that behaves like a basic List. Implement methods such as add(), remove(), get(), and size(). 9. Write a Java program to use lambda expressions for sorting a List of String objects. Demonstrate sorting by different criteria, such as string length and lexicographical order. 10. Write a Java program to use streams to perform complex operations on a collection. Create a list of objects, perform filtering, mapping, and reduction operations, and demonstrate the use of collect() to gather results.

References:

1. Schildt, H. *Java: The Complete Reference*. Tata McGraw-Hill Education, (2019)
2. Balagurusamy, E. *Programming with Java: A Primer*. Tata McGraw-Hill Education, (2014)
3. Kogent Solution Inc. *Java 6 Programming Black Book*. Dreamtech Press, (2007)
4. Java Official Documentation : https://docs.oracle.com/javase/tutorial/
5. Java Tutorial : https://www.geeksforgeeks.org/java/
6. Java Tutorial : https://www.javatpoint.com/java-tutorial

Program:	MCA						Semester : I
Course:	Python Programming Lab						Code : MCA31PC06
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
2	-	4	-	-	20	30	50

Prior knowledge of :

1. Computer Fundamentals
2. Basics Concepts of Programming

is essential.

Course Objectives

This course aims at enabling students:

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

Course Outcomes

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

1. Describe different concepts associated to Python programming.
2. Demonstrate the use of functions, modules
3. Apply the concepts of exception handling.
4. Perform different operations related to file handling using Python.
5. Implement different concepts of object oriented programming.
6. Perform CRUD Operations on MySQL database using Python.

Guidelines:

Students will be assessed based on

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

Students are advised to use:

A. Notebook
B. Python 3.X
C. MySQL WorkBench

Detailed Syllabus

Assign. No.	Suggested List of Assignments
1	Assignments based on control structures (If Else, Elif, For, While) 1. Create a Python program to find the Largest of Two Numbers: 2. Create a Python program to check if a number is prime or not.

	3. Use Python to create a simple calculator that can perform basic arithmetic operations. 4. Implement a Python function to find the largest element in a list. 5. Write a Python program that reads two numbers from the user, calculates their sum and difference, and prints the results with appropriate labels. 6. Create a Python script that converts a Fahrenheit temperature to Celsius and vice versa, using appropriate data types and conversion formulas. 7. Write a Python program that takes an integer input from the user and classifies it as positive, negative, or zero using if-elif-else statements. 8. Write a Python program to determine whether a given year is a leap year or not. Use conditional statements (if-else) to implement the logic. 9. write program on Grade Calculation Based on Marks 10. Create a Python program to check if a Number is Divisible by 5 and 3
2	Assignments based on input and outputs. 1. Develop a Python program that takes two numbers as input and performs addition, subtraction, multiplication, and division, printing each result. 2. Write a Python program that takes a string input from the user and prints the string in uppercase, lowercase, and with each word capitalized. 3. Create a Python program that asks the user for the radius of a circle and then calculates and prints the area and circumference. 4. Write a Python program that converts a temperature from Celsius to Fahrenheit and vice versa based on user input. 5. Write a Python program that asks the user for their name and a number, then prints the name repeated that many times. 6. Create a Python program that takes two numbers as input from the user and prints their sum. 7. Write a Python program that takes a string as input and prints the number of characters, words, and sentences in the string. 8. Develop a Python program that calculates simple interest. The user should input the principal, rate of interest, and time, and the program should output the interest amount. 9. Write a Python program that takes a sentence as input and prints each word in reverse order while keeping the word sequence the same. 10. Create a Python program that takes three numbers as input and prints the largest and smallest of the three.
3	Assignments based on Data Structures in Python (List, Tuple, Set, Dictionary). 1. Design a Python program to simulate a simple game of rock-paper-scissors. 2. Develop a Python code to generate a random password with specified criteria. 3. Create a Python script that calculates the area and perimeter of different geometric shapes based on user-provided dimensions. 4. Write a Python program that reads two numbers from the user, calculates their sum and difference, and prints the results with appropriate labels. 5. Create a Python script that converts a Fahrenheit temperature to Celsius and vice versa, using appropriate data types and conversion formulas. 6. Implement a Python function to sort a list of numbers using the bubble sort algorithm. 7. Write a Python program that takes a list of numbers as input and prints the sum of

	all the even numbers in the list. 8. Write a Python program that takes a list of strings and prints the count of each unique word in the list. 9. Write a Python program that defines two tuples of numbers and calculates the element-wise sum of the tuples. 10. Create a program that checks if a given element exists in a tuple and prints whether it is present or not.
4	Assignments based on Functions in Python 1. 1. Python Program to Find Numbers Divisible by Another Number 2. Python Program to Convert Decimal to Binary, Octal, and Hexadecimal 3. Python Program to Find ASCII Value of Character 4. Python program to find the H.C.F and L.C.M. of two input numbers 5. Python Program to Find Factors of Number 6. Python Program makes a simple calculator that can add, subtract, multiply and divide using functions 7. Python program to shuffle a deck of cards using the module random and draw 5 cards 8. Python program to display a calendar of a given month of the year 9. Python program to find the sum of natural numbers up to n using recursive function 10. Write a Python Program to demonstrate all the built-in functions from the Math module.
5	Assignments based on Functions in Python 2. 1. Create a Python function that calculates the area of a rectangle. The function should have default arguments for the width and height. Test the function with and without providing the arguments. 2. Develop a Python function that takes a list of numbers and returns the minimum, maximum, and average of the list. Call the function and print the results. 3. Create a Python function that accepts any number of arguments and returns their product. Test the function with different numbers of arguments. 4. Write a Python program that uses a lambda function to sort a list of tuples based on the second element of each tuple. Test the sorting function with sample data. 5. Develop a Python function that calculates the factorial of a number using recursion. Test the function with various inputs. 6. Write a Python function that takes a list of numbers as an argument and returns a new list with each element squared. Call the function and print the new list. 7. Create a Python function that takes a list of strings and converts each string to uppercase. The function should modify the original list. Test the function with a sample list. 8. Develop a Python function that checks whether a given number is prime. The function should return True if the number is prime and False otherwise. 9. Write a Python function that takes a dictionary as an argument and prints each key-value pair in the format key: value. Test the function with a sample dictionary. 10. Create a Python function that contains another function inside it. The inner function should perform a calculation, and the outer function should return the result of the inner function. Demonstrate the use of the nested function.

6	Assignments based on modules in Python 1. 1. Write a Python program to list all files and directories in the current working directory using the os module. 2. Create a Python script that accepts command-line arguments and prints the total number of arguments using the sys module. 3. Write a Python program that creates a directory, changes the current working directory to it, and then removes it using the os module. 4. Develop a script to fetch and display the current system path (sys.path) and environment variables using the osmodule.Write a Python program to find the size of a file in bytes using the os module. 5. Write a Python program to calculate the area of a circle using the math module. 6. Implement a Python function to compute the factorial of a number using the math module. 7. Write a Python program to generate a list of prime numbers up to a given number using the math.isqrt() method. 8. Create a Python function that calculates the square root of a number and rounds it to two decimal places using the math module. 9. Write a Python program to compute the greatest common divisor (GCD) of two numbers using math.gcd(). 10. Write a Python program to print the current date and time using the datetime module.
7	Assignments based on modules in Python 2. 1. Program to Work with Dates (Using datetime Module) 2. Program to Generate Random Numbers (Using random Module) 3. Program to Display Calendar of a Month (Using calendar Module) 4. Program to Calculate the Area and Circumference of a Circle 5. Program to Generate a Random Number Between Two Numbers 6. Program to Get Current Date and Time 7. Program to Create a Directory and List Files in it(Using the os module) 8. Program to Shuffle a List of Numbers(Using the random module) 9. Program to Display the Calendar for a Specific Month(Using the calendar module) 10. Program to Perform Arithmetic Operations Using a Custom Module 11. (Create a module math_operations.py with arithmetic functions)
8	Assignments based on Exception Handling in Python 1. 1. Write a Python program to input a number, if it is not a number generate an error message. 2. Write a Python program to catch ZeroDivisionError. Accept the values from the user. 3. Write a Python Program to Raise Custom Exception “PythonException” 4. Write a Python Program to demonstrate the use of try...except...else clause. 5. Write a Python Program to demonstrate the use of try...except...finally clause 6. Implement a program that handles both ValueError and TypeError exceptions when reading user input. 7. Create a custom exception class called NegativeNumberError to signal invalid input. 8. Write a function that raises a custom exception if a provided argument is not a valid file path. 9. Write a function that raises an InsufficientFundsError if a withdrawal amount exceeds the available balance. 10. Implement a program that raises a custom exception if a user enters an incorrect

	password multiple times.
9	Assignments based on Exception Handling in Python 2. 1. Create a Python library with custom exceptions for error handling. 2. Write a program that utilizes the else clause in a try-except block to perform cleanup tasks even if no exception occurs. 3. Implement a program that uses nested try-except blocks to handle specific exceptions within multiple layers of function calls. 4. Design a program that uses the finally clause to ensure resources are released regardless of exceptions or successful execution. 5. Write a program that uses exceptions to implement a retry mechanism for specific operations in case of temporary failures. 6. Write a Python program that prompts the user to enter a valid integer. Use a loop and exception handling to ensure that the user enters a valid integer, and print the entered integer. 7. Create a program that opens a file and reads its content. Handle the PermissionError exception and print a message if the program does not have permission to read the file. 8. Create a program that accepts user input for a list of numbers, and then calculates and prints the average of those numbers. Handle the ValueError exception when converting user input to numbers. 9. Create a program that asks the user to input two numbers and performs a mathematical operation (e.g., addition, subtraction, multiplication, division) based on user input. Handle the potential exceptions that may arise during user input or calculation. 10. Create a program that simulates a login system. Ask the user to enter a username and password. Handle the KeyError when checking if the entered username exists in a dictionary and handle the ValueError when comparing the entered password.
10	Assignments based on Input Output and File Handling using Python 1. 1. Write a Python program to read an entire text file. 2. Write a Python program to append text to a file and display the text. 3. Write a Python program to read a file line by line and store it into an array. 4. Write a python program to find the longest words. 5. Write a Python program to create a file where all letters of the English alphabet are listed by a specified number of letters on each line. 6. Implement a program that opens a file in write mode, but only if it doesn't already exist. 7. Write code to read a CSV file containing product information and create a list of product objects. 8. Create a program that generates a CSV report from a list of dictionaries. 9. Write a Python program that reads the contents of a text file and prints the number of lines, words, and characters in the file. 10. Create a program that takes user input to write a series of lines to a new text file. Ensure that each line is written to the file on a new line.
11	Assignments based on Input Output and File Handling using Python 2. 1. Write a Python program that reads a CSV file containing student names and their

	corresponding scores. Calculate the average score and print the names of students with scores above the average. 2. Write a Python program that appends a timestamp to each line of an existing text file, indicating when each line was last modified. 3. Create a program that reads a binary file and counts the occurrences of a specific byte pattern. Handle the IOError exception and print a message if there is an issue with file I/O. 4. Write a Python program that reads a JSON file containing a list of books. Allow the user to search for a book by title and print its details if found. 5. Write a Python program that creates a new directory and then creates multiple text files inside that directory, each with a different name and content. 6. Create a program that reads a log file and extracts information about errors. Print the count of each type of error along with the corresponding error messages. 7. Create a program that reads a text file containing a list of email addresses. Identify and print all unique domains present in the email addresses. 8. Write a Python program that reads a binary file and performs encryption on its contents. Save the encrypted data to a new file. 9. Write a Python program that accepts user input for a file path and checks whether the file exists. If it does, print the file's size; otherwise, print a message indicating that the file doesn't exist. 10. Create a program that reads a JSON file containing information about employees. Allow the user to add a new employee to the file and save the updated data.
12	Assignments based on Object Oriented Programming 1. 1. Create a Vehicle class with max_speed and mileage instance attributes and print max_speed and mileage. 2. Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number. 3. Create a Bus child class that inherits from the Vehicle class. The default fare charge of any vehicle is seating capacity * 100. If Vehicle is a Bus instance, we need to add an extra 10% on the full fare as a maintenance charge. So total fare for bus instance will become the final amount = total fare + 10% of the total fare. 4. Write a Python class to convert an integer to a roman numeral. 5. Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' and ']. These brackets must be close in the correct order, for example "()" and "()[]{}" are valid but "[)", "({[)]" and "{{{" are invalid. 6. Write a Python program that defines a Car class. Include attributes like make, model, and year. Create an object of the class and print its details. 7. Create a program that defines a Student class. Include attributes for name, age, and grade. Write a method to display the student's information. 8. Write a Python program that uses a constructor to initialize an object of a Person class with a given name and age. Print the details of the person. 9. Create a program that defines a Book class with a constructor that sets the book title and author. Use a destructor to print a message when an object is destroyed. 10. Write a Python program that demonstrates inheritance. Define a base class Animal with a method make_sound(). Create derived classes Dog and Cat that override the method.

13	Assignments based on Object Oriented Programming 2. 1. Create a program that models a basic shape hierarchy. Define a base class Shape with methods for calculating area and perimeter. Create subclasses for Circle and Rectangle. 2. Write a Python program that demonstrates polymorphism. Define a class Shape with a method draw(). Create subclasses Circle and Square that implement the draw() method. 3. Create a program that uses polymorphism to calculate the total cost of items in a shopping cart. Define a base class Item and subclasses for different types of items. 4. Write a Python program that defines an abstract class Shape with abstract methods for calculating area and perimeter. Create subclasses Circle and Rectangle to implement these methods. 5. Create a program that defines an interface Payment with methods for processing payments. Implement the interface in classes CreditCardPayment and PayPalPayment. 6. Write a Python program that models a simple bank account hierarchy. Create a base class Account with attributes for account_number, account_holder, and balance. Create subclasses for SavingsAccount and CurrentAccount. 7. Create a program that simulates a zoo. Define a base class Animal with attributes for name and sound. Create subclasses for different types of animals (e.g., Lion, Elephant) that inherit from the Animal class. 8. Create a program that models different types of vehicles. Define a base class Vehicle with methods for start_engine() and stop_engine(). Create subclasses for Car and Motorcycle that implement these methods. 9. Write a Python program that simulates a university course registration system. Define a base class Course with methods for enrolling and dropping students. Create subclasses for different types of courses (e.g., MathCourse, HistoryCourse). 10. Create a program that models a shape drawing application. Define a base class DrawingTool with methods for drawing shapes. Create subclasses for different types of drawing tools (e.g., PenTool, BrushTool) with specific drawing behavior.
14	Assignments based on Database Handling 1. 1. Write a Python program to create a table and insert some records in that table. Finally selects all rows from the table and display the records 2. Write a Python program to update a specific column value of a given table and select all rows before and after updating the said table. 3. Write a Python program to delete a specific row from a given table. 4. Write a Python program to alter a given table. 5. Write a Python program to update all the values of a specific column of a given table. 6. Write Python code to: a. Connect to a MySQL database using the mysql.connector library. b. Execute a SQL query to retrieve all customer names and emails. c. Insert a new row into the "customers" table. d. Update an existing customer's email address. 7. Write a Python program that connects to a MySQL database and fetches and prints the version of the MySQL server.

	8. Create a program that establishes a connection to a MySQL database and lists all the databases available on the server. 9. Write a Python program that connects to a MySQL database, creates a table named employees with columns for id, name, and salary, and inserts a few records into the table. 10. Create a program that connects to a MySQL database, queries the employees table, and prints the names and salaries of employees
15	Assignments based on Database Handling 2. 1. Write a Python program that connects to a MySQL database, updates the salary of an employee in the employees table, and then queries and prints the updated record. 2. Create a program that connects to a MySQL database, deletes a specific employee record from the employees table, and then queries and prints the remaining records. 3. Write a Python program that demonstrates the use of transactions. Connect to a MySQL database, start a transaction, insert multiple records into a table, and commit the transaction. 4. Create a program that connects to a MySQL database, starts a transaction, updates the salary of an employee, but rolls back the transaction if an error occurs during the update. 5. Write a Python program that connects to a MySQL database, queries a table containing student grades, and calculates and prints the average grade. 6. Create a program that connects to a MySQL database, queries a table containing customer orders, and prints the total revenue for a specific product. 7. Write a Python program that connects to a MySQL database and uses a prepared statement to insert multiple records into a table. 8. Create a program that connects to a MySQL database, uses a prepared statement to update the salary of an employee, and then queries and prints the updated record. 9. Write a Python program that connects to a MySQL database, creates two tables named students and courses, establishes a relationship between them, and inserts sample data. 10. Create a program that connects to a MySQL database, performs a JOIN operation on the students and courses tables, and prints the names of students along with the courses they are enrolled in.

References:

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

Program:	**MCA**	**Semester : I**
Course:	**Research Methodologies and IPR**	**Code : MCA31EL01**

Credits	**Teaching Scheme (Hrs. / Week)**				**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**O***	**FA**		**TW**	**SA**	**Total**
					FA-1	**FA-2**			
4	**3**	**-**	**1**	**6**	**20**	**20**	**-**	**60**	**100**

Prior knowledge of :

1. Fundamentals of Computer
2. Basics of Business Statistics

is essential.

Course Objectives

This course aims at enabling students:

1. To develop an understanding of various research types and methodologies.
2. To identify various sources of information for literature review and data collection.
3. To learn the literature review concepts, technical reading, writing, and citations.
4. To develop an understanding of research and publication ethics.
5. To understand the concepts of intellectual property rights.
6. To learn the procedure of obtaining patents, copyrights, trademarks and industrial design

Course Outcomes

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

1. Understand the research methodology and explain the technique of defining a research problem.
2. Explain the details of sampling designs and different data collection methods.
3. Search, select, and critically analyze research articles and papers.
4. Apply the appropriate statistical methods required for research design and analysis.
5. Explain various forms of intellectual property, their relevance, rights, and business impact in the changing global business environment
6. Identify procedures to protect different forms of IPRs at national and international levels.

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Introduction to Research Methodology** Meaning of Research, Objectives of Research, Motivation in Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, Problems Encountered by Researchers in India	6
2	**Research and Sampling Design** What is a Research Problem?, Selecting the Problem, Important Concepts Relating to Research Design, Census, and Sample Survey, Implications of a Sample Design, Steps in Sampling Design, Criteria of Selecting a Sampling Procedure, Characteristics of a Good Sample Design, Different Types of Sample Designs, Examples	6

3	**Literature Review, Technical Reading, Writing and Citations** New and Existing Knowledge, Analysis and Synthesis of Prior Art Bibliographic Databases, Science Citation Index (SCI), Web of Science, Scopus, IEEE Xplore, and Google Scholar, Effective Search: The Way Forward Introduction to Technical Reading Conceptualizing Research, Critical and Creative Reading, Taking Notes While Reading, Reading Mathematics and Algorithms, Reading a Datasheet. Citations: Functions and Attributes, Knowledge Flow through Citation, Citing Datasets, Styles and Tools for Citations, Research Acknowledgments.	10
4	**Data Collection, Processing, and Analysis of Data:** Collection of Primary Data, Collection of Data through Questionnaires, Collection of Secondary Data, Selection of Appropriate Method for Data Collection, Processing Operations, Elements/Types of Analysis, Statistics in Research, Measures of Central Tendency, Measures of Dispersion, Measures of Asymmetry (Skewness), Measures of Relationship, Simple Regression Analysis, Multiple Correlation and Regression Analysis, other Measures, and Examples	8
5	**Introduction to Intellectual Property Rights** Role of IP in the Economic and Cultural Development of the Society, IP Governance, IP as a Global Indicator of Innovation, Categories of Intellectual Property: Patents, Copyrights and Related Rights, Trademarks, Geographical Indications, Trade Secrets, Semiconductor Integrated Circuits and Layout Designs, Plant Varieties, Traditional Knowledge, Industrial Designs, Biodiversity Conservation, Major Amendments in IP Laws and Acts in India	8
6	**IP Ecosystem in India, and Introduction to WIPO** Department for Promotion of Industry and Internal Trade (DPIIT), National Research Development Corporation (NRDC), Technology Information Forecasting and Assessment Council (TIFAC), Patent Facilitation Centre, Indian Web Portals for Patents and Technologies, Origin of WIPO, Salient Features of WIPO, Main Activities of WIPO, India and WIPO**.**	7
Other*	Use of Mendeley and Overleaf citation management tools. Study of various citation matrices and DOI. Study of Shodhganga and Shodhgangotri web portals. Explore the process of obtaining a patent or copyrights, the different types, and the rights and obligations associated with it.	6
	Total	**45+6=51**

List of Tutorial / Activities:

1. Finding and using primary and secondary sources for research papers and data collection.
2. Writing systematic literature reviews based on the existing research papers.
3. Recognizing statistics, facts, arguments, and opinions in the existing research literature.
4. Identifying and writing the research gaps in the existing research literature.
5. Developing a research question, choosing a research topic, and writing the aims and objectives of the research while working collaboratively with others.
6. Writing technical analysis and citing the research using various citation tools.
7. Recognizing ethical issues, understanding legal requirements, avoiding plagiarism, and improving the quality of research.
8. Research paper presentation/submission in quality national or international journals/conferences.
9. Develop an understanding of India's intellectual property rights and registration process.
10. Identify the role of WIPO (World Intellectual Property Organization) in developing IPR.
11. Identify the emerging trends in intellectual property law in India.

12. Presentation of technical ideas in posters, papers, copyrights, trademarks, patents, etc.

Text Books:

1. C.R. Kothari, GauravGarg, *Research Methodology: Methods and Techniques*,, New Age International, 2020
2. Prof. RupinderTewari Ms. MamtaBhardwa, *Intellectual Property A Primer for Academia (https://dst.gov.in/sites/default/files/E-BOOK%20IPR.pdf)*

Reference Books:

1. Ranjit Kumar, *Research Methodology: A step-by-step guide for beginners*, Sage Publications, 2023.
2. Naval Bajpai, *Business Statistics*, Pearson Publication. 2010.
3. Deborah E. Bouchoux, *Intellectual Property: The Law of Trademarks, Copyrights, Patents, and Trade Secrets*, 2012.

E-Resources:

1. NOC: Research Methodology, https://nptel.ac.in/courses/127106227
2. NOC: Research Methodology and IPR, https://onlinecourses.swayam2.ac.in/ntr24_ed08/preview
3. Official Website of Intellectual Property India, https://www.ipindia.gov.in/
4. Official Website of World Intellectual Property Organization (WIPO), https://www.wipo.int/portal/en/index.html

Program:	MCA	Semester : I
Course:	Mathematical Foundation for Computer Application-1	Code : MCA31BS01

Credits	Teaching Scheme (Hrs. / Week)				Evaluation Scheme and Marks				
	Lecture	Practical	Tutorial / Activity	O*	FA		TW	SA	Total
					FA-1	FA-2			
4	3	-	1	6	20	20	-	60	100

Prior knowledge of :

1. Basic Mathematics

is essential.

Course Objectives

This course aims at enabling students:

1. To recognize the mathematical underpinnings of probability.
2. To study decisions about likelihood of events, based on a pattern of collected data.
3. To recognize and understand probability distribution function

Course Outcomes

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

1. Use probability theory to solve permutation and combination problems.
2. Apply the Addition Rule and the Principle of Inclusion and Exclusion.
3. Implement the concept of Conditional Probability and probability.
4. Analyze the discrete probability distribution and apply it appropriately.
5. Analyze the continuous and normal probability distribution and apply it appropriately

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Counting Principle** Addition and Multiplication Principles, Permutations of n Objects, Circular Permutation, Permutation with repetitions.	7
2	**Principle of Inclusion and Exclusion** Principle of Inclusion and Exclusion theorem and applications, Derangement theorem and its applications, on negative integer value solution, Multinomial Theorem and application.	7
3	**Probability** Trail,Events,Sample spaces, probability axioms, Independent and Dependent Events, Conditional probability and its applications,Bayes's Theorem and its applications.	6
4	**Random variables and Mathematical Expectation** Random Variable (Discrete and continuous),Probability Distribution of a Random Variable, Probability Mass Function, Probability Density Function, Distribution Function., Mathematical Expectation of Probability Distribution, Theorems, Calculation of Mean and Variance using Mathematical Expectation, Concepts of Bivariate Random Variable, Discrete and Continuous Bivariate Random Variable	9

5	**Discrete Probability Distribution:** Binomial Distribution, Finding Mean and variance of Binomial Distribution , Poisson Distribution , Finding Mean and variance of Poisson Distribution, Numerical base on Binomial Distribution and Poisson Distribution	8
6	**Continuous Probability Distribution:** Uniform Distribution, Finding Mean and variance of uniform DistributionNormal Distribution ,Numerical base on Uniform Distribution and Normal Distribution	8
Other*	Applicability of Bayes theorem, Poisson Distribution, Uniformdistribution, Normal distribution	6
	Total	**45+6=51**

List of Tutorial / Activities

1. Activity on Counting Principle
2. Activity on Inclusion and Exclusion
3. Activity on Probability
4. Activity on Mathematical Expectation
5. Activity on Discrete Probability distribution
6. Activity on Continuous Probability Distribution

Text Books:

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

Reference Books:

1. Williams, A. S. W. A. S. *Statistics for business and economics*. South-Western,A TRIZ. 2011.
2. Judge, G. G., Griffiths, W. E., Hill, R. C., Lütkepohl, H., & Lee, T. C. *The theory and practice of econometrics* . John Wiley & Sons.1991.
3. Gupta, S. P. *Statistical Methods By SP Gupta*. Sultan Chand and Sons, Delhi, India.2001.

E-Resources:

1. Statistical Concept:Statistical Methods calculators (atozmath.com)
2. Discrete Mathematics: https://www.tutorialspoint.com/discrete_mathematics/index.htm

Syllabus of Courses
Semester II
First Year MCA

Program:	**MCA**	**Semester : II**
Course:	**Operating System Concepts**	**Code : MCA32PC07**

Credits	**Teaching Scheme (Hrs. / Week)**				**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**O***	**FA**		**TW**	**SA**	**Total**
					FA-1	**FA-2**			
2	**2**	**-**	**-**	**4**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. Computer Fundamentals

is essential.

Course Objectives

This course aims at enabling students:

1. To introduce fundamental ideas and capabilities of contemporary operating systems.
2. To comprehend the idea of process and thread management.
3. To comprehend different memory management strategies.

Course Outcomes

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

1. Identify knowledge of the role of Operating Systems and their types.
2. Apply the concept of a process, thread and scheduling algorithms.
3. Use process synchronization and deadlock problems related to inconsistency and race conditions with shared variables.
4. Demonstrate various memory management techniques

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Operating System Overview:** Operating System Objectives and Functions, The Evolution of Operating Systems, Developments Leading to Modern Operating Systems, Virtual Machines, OS Design Considerations for Multiprocessor and Multicore architectures, Microsoft Windows Overview, Modern UNIX Systems, Linux, Android. Booting Process of all the above operating systems.	5
2	**Process:** Concept of a Process, Process States, Process Description, Process Control Execution of the Operating System. Threads: Processes and Threads, Concept of Multithreading, Types of Threads, Thread Management Scheduling: Types of Scheduling, Scheduling Algorithm	9
3	**Concurrency:** Mutual Exclusion and Synchronization Concurrency: Process/thread Synchronization and Mutual Exclusion Principles of Concurrency, Requirements for Mutual Exclusion, Mutual Exclusion: Hardware Support, Operating System Support ,Classical synchronization problems: Readers/Writers Problem, Producer and Consumer problem. Concurrency: Principles of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock detection	9

4	**Memory Management:** Background ,Swapping ,Contiguous Memory Allocation ,Paging , Segmentation, Virtual Memory , Demand Paging , Page Replacement	7
Other*	Study of different Operating, Systems(Linux, Windows, Android OS, iOS) Shell Scripting	4
	Total	**30+4=34**

Text Books:

1. William Stallings, *Operating System: Internals and Design Principles*, Prentice Hall, 2014.
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, *Operating System Concepts*, John Wiley &Sons ,Inc.2012.

Reference Books:

1. Tom Adelstein and Bill Lubanovic, *Linux System Administration*, O'Reilly Media, Inc.2007.
2. Harvey M. Deitel, *Operating Systems,* Prentice Hall,2003,
3. Andrew S. Tanenbaum, *Modern Operating System*, Prentice Hall, 2007.

E-Resources:

1. To Understand Operating System Basics: https://nptel.ac.in/courses/106/105/106105214/
2. Unix:https://www.youtube.com/watch?v=1hf_0EeOYBY (NPTEL IIT Madras)

Program:	**MCA**						**Semester : II**
Course:	**Web Technologies Lab**						**Code : MCA32PC08**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Computer Fundamentals
2. Object Oriented Programming

is essential.

Course Objectives

This course aims at enabling students:

1. To acquire the knowledge of HTML, CSS, Bootstrap and JavaScript.
2. To gain the skills needed for entry into web application and development careers.

Course Outcomes

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

1. Use HTML elements to create the web pages
2. Style and/or layout the web pages using CSS
3. Build responsive web pages using Bootstrap
4. Develop interactive web pages using basics of JavaScript
5. Use advanced JavaScript features to build dynamic web pages

Guidelines:

Students will be assessed based on:

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

Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks**.**

Students are advised to use:

1. Version control system such as Git
2. Free Web Hosting such as Netlify, Herokuetc

Use of VS Code editor is recommended. Students may also use alternatives such as Atom, Sublime.

Detailed Syllabus

Assign No.	**Suggested List of Assignments**
1	**Case Study 1: E-Commerce Product Page Analysis** **Scenario:** A startup wants to launch an e-commerce website. They require a product page that effectively displays product details, images, and user reviews. **Assignment Tasks:** 1. Analyze the structure and layout requirements for an e-commerce product page.

	2. Identify appropriate HTML elements for product descriptions, images, and user-generated reviews. 3. Evaluate the use of semantic tags to enhance the page's SEO and accessibility. 4. Create a wireframe and justify your layout choices. 5. HTML (Web Content Definition) a. Define the structure of the product page, including sections for product details, images, and reviews. b. Use semantic elements for better organization. 6. CSS (Styling) a. Style the product page with color schemes, fonts, and layout improvements. b. Ensure proper alignment and visual consistency. 7. Bootstrap (Responsiveness) a. Implement a responsive grid layout using Bootstrap. b. Ensure the page displays well on different screen sizes. 8. JavaScript (Validation & Feedback) a. Add form validation for user reviews or inquiries. b. Display a success message when a form is submitted correctly.
2	**Case Study 2: News Website Homepage** Scenario: A media company plans to revamp the homepage of its news portal. They aim to improve the layout and accessibility for a better user experience. Assignment Tasks: 1. Analyze the content distribution and visual hierarchy of existing news websites. 2. Propose an HTML structure for the homepage that includes categories for the latest news, editor picks, trending topics, etc. 3. Justify your choice of HTML5 semantic tags and their relevance for a news website. 4. HTML (Web Content Definition) a. Define sections for the latest news, editor picks, trending topics, etc, using semantic HTML elements. 5. CSS (Styling) a. Style the homepage with fonts, colors, and visual hierarchy improvements. b. Create a visually appealing header and footer. 6. Bootstrap (Responsiveness) a. Use Bootstrap grid classes to make the homepage fully responsive. 7. JavaScript (Validation & Feedback) a. Implement a search box validation to ensure users enter a keyword before searching. b. Provide feedback on search results or error handling.

References:

1. Powel T. A., *HTML and CSS – The Complete Reference*, McGraw Hill, 2018
2. Arrardi J., *Modern CSS,* Apress Publication, 2020
3. Complete Bootstrap : bootstrap.com
4. Javascript Basic to Advanced : javascript.info
5. Javascript Basic to Advanced : javascripttutorial.net

Program:	**MCA**						**Semester : II**	
Course:	**Data Warehouse and Data Mining (PEC-1)**						**Code : MCA32PE01**	
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. DBMS
2. Data Structure

is essential.

Course Objectives

This course aims at enabling students:

1. To understand fundamental concepts, techniques and design principles of data warehousing and data mining.
2. To enable students to understand, implement and evaluate various algorithms in data mining

Course Outcomes

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

1. Apply different preprocessing methods to prepare data in the desired format.
2. Create Data Warehouse schemas and apply OLAP operations for effective multidimensional data analysis.
3. Evaluate frequent patterns using Association Mining Techniques on large datasets.
4. Evaluate various Clustering and Classification Techniques on large datasets

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Data Pre-processing** Data Objects attribute types, descriptions of data, Measuring Data Similarity and Dissimilarity on binary, numerical and mixed dataset. Data Pre-processing. Introduction to weka./Tanagra /Orange tool for data preprocessing	6
2	**Dimensional Modeling** Data Warehouse Schemas: Star Schema, Snowflake Schema and Fact Constellation Schema. OLAP and operations on Multidimensional Database: Rollup, Roll down, Dice, Slice and Pivot.	6
3	**Introduction to Data Mining and Association Rules** Concept of Data Mining, KDD, Applications of Data Mining. Define Association Rule, Apriori Algorithm and Frequent-pattern Tree Algorithm to find frequent item set and strong association rules.	8
4	**Clustering and Classification** Introductions to Clustering and Classification, Input and Output Attributes, Naïve Bayes Classification, k-Nearest-Neighbor Classifiers (Lazy Learners),	10

	Clustering: Major Clustering Algorithms-Partition Clustering: k-means clustering, Issues with the k-means algorithm, Hierarchical clustering: Agglomerative clustering and Divisive clustering. A case study on finding efficient Clusters/classification on sample data set.	
	Total	**30**

Text Books:

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

Reference Books:

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

E-Resources:

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

Program:	**MCA**	**Semester : II**
Course:	**Web Development using Django (PEC-1)**	**Code : MCA32PE02**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. Basic Programming Concepts
2. Basics of Web Development
3. Logical Thinking
4. Database Operations

is essential.

Course Objectives

This course aims at enabling students:

1. Understand the concept of dynamic web pages and the Django template system.
2. Learn how to map URLs to views and handle 404 errors in Django.
3. Gain knowledge of the Model-View-Template (MVT) development pattern in Django.
4. Develop skills in defining and installing database models using Django's ORM.
5. Explore the Django admin interface and its customization capabilities.
6. Learn about form processing, validation, and saving form responses in Django.
7. Understand access control mechanisms using sessions and user authentication in Django.
8. Implement login, logout, user management, and permission handling in Django applications.
9. Gain exposure to asynchronous messaging concepts

Course Outcomes

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

1. Create dynamic web pages using Django's template system and context objects.
2. Implement database models.
3. Implement Django admin interface for efficient management of application data, including users, groups, permissions, and form processing.
4. Implement the Django session framework to manage user sessions, handle session data in views, optimize session usage for performance, and ensure secure session management practices.

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Dynamic Web Pages And Django Template System :** Dynamic Content, Mapping URLs to Views, URL configurations and Loose Coupling, 404 Errors, Dynamic URLs, Django Template System Basics, Creating Template Objects, Rendering a Template, Rendering a Template, Playing with Context Objects	8
2	**Database Model With Django :**The MVT Development Pattern, Configuring Django for Database Access, Understanding Django Apps, Defining Models in Python, Installing the Model, Understanding Model	7

	Fields & Options, Creating A Django Model, Adding Model String Representations, Making Changes to a Database Schema.	
3	**Django Admin Interface :** Activating the Admin Interface, Using the Admin Interface, Users, Groups and Permissions Customizing the Admin Interface's Look and Feel, Customizing the Admin Index Page, When and Why to Use the Admin Interface, Form Processing, Form Validation Save Form Response	8
4	**Access Control With Session And Users :** The Django Session Framework, Sessions in Views, Session Tuning, Installing Django User Authentication, Using Authentication in Views, Login and Logout, Building your Own Login/Logout Views, Adding & Deactivating Users, Asynchronous Messaging, Managing Permissions	7
	Total	**30**

Text Books:

1. George, N. *Build a Website with Django 3*. Packt Publishing, (2020)

Reference Books:

1. Greenfeld, D. R., &Greenfeld, A. R. *Two Scoops of Django 1.11: Best Practices for the Django Web Framework*. Two Scoops Press, (2017)
2. Vincent, W. S. *Django for Beginners: Build Websites with Python and Django. WelcomeToCode*, (2021)

E-Resources:

1. **Django Tutorial :** https://www.w3schools.com/django/
2. **Learn Django Framework :** https://www.geeksforgeeks.org/django-tutorial/
3. **Django Tutorial :** https://www.tutorialspoint.com/django/index.htm

Program:	**MCA**	**Semester : II**
Course:	**UI/UX Design (PEC-1)**	**Code : MCA32PE03**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. HTML
2. CSS
3. Understanding of Basic Design Principles

is essential.

Course Objectives

This course aims at enabling students:

1. Understand the principles and importance of user-centered design.
2. Learn essential UI design principles and tools.
3. Develop proficiency in UX research methods and techniques.
4. Gain hands-on experience in prototyping, wireframing, and usability testing.
5. Explore advanced UI design concepts and emerging trends in the field.

Course Outcomes

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

1. Apply user-centered design principles to create intuitive and user-friendly interfaces.
2. Design visually appealing and functional UI elements using industry-standard tools.
3. Conduct effective user research and apply findings to inform design decisions.
4. Develop prototypes and wireframes to visualize and test interface designs.
5. Analyze and optimize user experiences through usability testing and iteration.

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Introduction to UI/UX Design** **Understanding User-Centered Design :** Introduction to User Experience (UX) and User Interface (UI) design principles, Importance of empathy in design: understanding user needs, goals, and pain points, Overview of the design thinking process. **UI Fundamentals : Basic principles of UI design:** layout, typography, color theory, and visual hierarchy, Introduction to design tools such as Adobe XD, Sketch, or Figma. **UX Research Methods:** Overview of user research techniques: surveys, interviews, usability testing, and analytics, Creating user personas and user journey maps, Conducting usability tests and analyzing results. **Prototyping and Wireframing:** Introduction to wireframing and prototyping tools, Creating low-fidelity wireframes to outline page layouts and functionality, Prototyping interactive mockups to simulate user interactions.	7

2	**Advanced UI Design** **Responsive Design and Mobile UI :** Principles of responsive web design: fluid grids, flexible images, and media queries, Designing for different screen sizes and devices, Mobile UI best practices and common design patterns. **Advanced UI Elements:** Designing complex UI components: dropdowns, sliders, tabs, etc., Animation and micro-interactions to enhance user experience, Accessibility considerations in UI design. **Visual Design Principles :** Advanced color theory and psychology in UI design, Gestalt principles and how they apply to UI design, Creating visually engaging UIs through use of whitespace, contrast, and visual balance **UI Design Trends and Future Directions :** Exploration of current UI design trends, Predicting future directions in UI design, Case studies of innovative UI designs and their impact.	8
3	**UX Design Process** **Information Architecture :** Principles of organizing and structuring information, Creating sitemaps and navigation systems, Card sorting techniques for information organization **Interaction Design :** Principles of interaction design, feedback, affordances, and signifiers, Mapping user flows and designing intuitive interactions, Implementing user feedback loops for iterative design. **Usability Testing and Evaluation :** Planning and conducting usability tests, Analyzing usability metrics and identifying areas for improvement, Iterative design based on user feedback. **UX Design Documentation :** Creating design documentation: wireframes, user flows, and prototypes, Presenting design rationale and justifying design decisions, Collaborating with developers and other stakeholders.	8
4	**Special Topics in UI/UX Design** **Design Systems and Component Libraries :** Introduction to design systems and their benefits, Creating and maintaining design systems, Integrating design systems into the design workflow **Cross-Cultural Design Considerations :** Understanding cultural differences in design preferences and usability, Adapting designs for international audiences, Case studies of successful cross-cultural design implementations. **Ethical Considerations in UI/UX Design :** Ethics in design: privacy, accessibility, and inclusivity, Designing ethically responsible products and experiences, Avoiding dark patterns and manipulative design practices.	7
	Total	**30**

Text Books:

1. Krug S., *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*, New Riders Publication, 2014
2. Norman D., *The Design of Everyday Things*, Basic Books Publication, 2013

Reference Books:

1. Garrett J. J., *The Elements of User Experience: User-Centered Design for the Web and Beyond*, New Riders Publication, 2010
2. Knight W., *UX for Developers*, Apress Publication, 2018

E-Resources:

1. UI/UX and Figma : https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-with-wireframe-prototype-figma/
2. UI/UX Tutorial : https://intellipaat.com/blog/ui-ux-design-tutorial/

Program:	**MCA**						**Semester : II**	
Course:	**Start-Up and New Venture Development (PEC-1)**						**Code : MCA32PE04**	
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :
No Prior knowledge is required

Course Objectives
This course aims at enabling students:

1. To develop entrepreneurship awareness
2. To inculcate an entrepreneurial mindset into the minds of young professionals
3. To identify entrepreneurial opportunities
4. To leverage skills for founding, leading & managing Startups

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

1. Develop traits and factors influencing the development of entrepreneurship as a profession
2. Develop a strategic plan for launching a Startup
3. Discover skill sets required for successful Entrepreneurship
4. Summarize the role of Government's various support organizations in promoting Entrepreneurship

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Introduction to Entrepreneurship** Why to become an entrepreneur ,Types of Entrepreneurs,Role of Entrepreneurship in Economic Development,Factors affecting Entrepreneurship,Skill Sets required for Successful Entrepreneurs	7
2	**Introduction to Startup Management** Concept of Start-Up,The Rise of The Startup Economy,Start Up India Policy ,Registration and Legal Process of Startups.Types of Business Organizations -Sole Proprietorship, Partnership, Limited Liability Partnership (LLP), Private Limited Company	7
3	**Skills for Successful Entrepreneurs** Communication Skills,Creativity and Innovation,Negotiation Skills Risk management,Case Study of Successful Entrepreneurs	8
4	**Role of Government and Institutional Support** Role of Government in promoting Entrepreneurship,MSME policy in India, Start-up India, Make in India schemes,Various Government Schemes Institutional Support for Entrepreneurship	8
	Total	**30**

Text Books:

1. Desai, V. *Dynamics of Entrepreneurship Development.* Himalaya Publishing House, (2017).
2. Holt, D. H. *Entrepreneurship: New Venture Creation.* Prentice Hall, (2019).

Reference Books:

1.Taneja, S., Gupta, S.L. *Entrepreneurship Development New Venture Creation. S*. Chand & Company Ltd, (2015).

2. Charantimath, P. M. *Entrepreneurship Development and Small Business Management*. Pearson Education India, *(2016).*

E-Resources:

1. https://www.startupindia.gov.in/
2. https://www.aim.gov.in/
3. https://www.investindia.gov.in/
4. https://msme.gov.in/
5. https://www.sidbi.in/

<table>
<tr><td><b>Program:</b></td><td colspan="6"><b>MCA</b></td><td colspan="2"><b>Semester : II</b></td></tr>
<tr><td><b>Course:</b></td><td colspan="6"><b>Data Science (PEC-2)</b></td><td colspan="2"><b>Code : MCA32PE09</b></td></tr>
<tr><td rowspan="3"><b>Credits</b></td><td colspan="3"><b>Teaching Scheme (Hrs. / Week)</b></td><td colspan="5"><b>Evaluation Scheme and Marks</b></td></tr>
<tr><td rowspan="2"><b>Lecture</b></td><td rowspan="2"><b>Practical</b></td><td rowspan="2"><b>Tutorial / Activity</b></td><td colspan="2"><b>FA</b></td><td rowspan="2"><b>TW</b></td><td rowspan="2"><b>SA</b></td><td rowspan="2"><b>Total</b></td></tr>
<tr><td><b>FA-1</b></td><td><b>FA-2</b></td></tr>
<tr><td><b>2</b></td><td><b>2</b></td><td><b>-</b></td><td><b>-</b></td><td><b>10</b></td><td><b>10</b></td><td><b>-</b></td><td><b>30</b></td><td><b>50</b></td></tr>
</table>

Prior knowledge of :

1. Basics of Python Programming
2. Basics of Business Statistics

is essential.

Course Objectives

This course aims at enabling students:

1. To develop relevant Python programming abilities for analysis of data.
2. To learn and demonstrate the basic statistical analysis of data.
3. To understand the features and applications of machine learning algorithms.
4. To use appropriate tools and methods to collect, process, summarize, and visualize data for analysis.

Course Outcomes

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

1. Understand the basics concepts of Data Science.
2. Apply basic statistical operations using Python programming libraries on datasets.
3. Analyse and implement data visualization techniques for data analysis.
4. Analyse and implement machine learning techniques for data analysis and visualization.

Detailed Syllabus

Unit No.	Description	Duration (Hrs.)
1	**Introduction to Data Science** Basic Concepts and Definitions, Big Data in the Enterprise, Data Science Industry Use Cases, Realizing Opportunities from Data, New Revenue Growth Opportunities and Applications, Data Analytics and Various Types	7
2	**Statistics for Data Science** Introduction to Business Statistics, Scales of Measurement, Qualitative and Quantitative data, Types of Statistics: Descriptive and Inferential, Measures of Central Tendency: Mean, Median, Mode, Percentiles, Quartiles, Measures of Variations: Range, Interquartile Range, Variance, Standard Deviation, Measures of shape- Skewness, Kurtosis, Introduction to Charts and Graphs	7
3	**Python Libraries** Introduction to Python Libraries, installing and importing, NumPy - Numerical Python Library: Arrays – Important Characteristics and Attributes, NumPy Array vs Python List, Creation of NumPy Arrays from	8

	List, Intrinsic Creation of an Arrays, Indexing, Slicing and Iterating an array, Basic Statistical Operations on Arrays, Broadcasting, Loading and Saving Arrays in Files, Pandas - Python Library, pandas Data Structures: Series,Dataframes, Defining a Series from NumPy Arrays and Other Series, Operations and Mathematical Functions on Series, Defining a dataframe: DataFrame() constructor, Assigning Values, Indexing, Membership, NaN Values, Operations Between DataFrame and Series, pandas: Reading and Writing Data using files	
4	**Data Analysis and Visualization Using Machine Learning Algorithms** The matplotlib Python Library, The matplotlib Architecture, A Simple Interactive Chart, Working with Multiple Figures, Elements and Axes, Charts with pandas: Line Charts, Histograms, Bar Charts, Pie Charts, Scatter Plots, etc., Advanced Charts: Contour Plots, Polar Charts, Plots in 3D, Working with subplots and grids, Examples using datasets, Regression: Linear and Multiple, Classification: Logistic and Decision Tree	8
	Total	**30**

Text Books:

1. Fabio Nelli, *Python for Data Analytics with Pandas, Numpy, and MatPlotLib*, Apress Publication, 2023.

Reference Books:

1. Wes McKinney, *Python for Data Analysis*, O'Reilly Publication, 2017.
2. Naval Bajpai, *Business Statistics*, Pearson Publication. 2010.

E-Resources:

1. NOC: Python for Data Science, https://nptel.ac.in/courses/106106212
2. NOC: Introduction to Data analytics, https://nptel.ac.in/courses/110106072

Program:	**MCA**	**Semester : II**
Course:	**Cloud Computing (PEC-2)**	**Code : MCA32PE10**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

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

is essential.

Course Objectives

This course aims at enabling students:

1. To provide an overview of cloud computing, encompassing its core principles, technologies, challenges, and real-world applications.
2. To understand the virtualization technologies and architectural concepts.
3. To explore the interconnectedness between cloud computing and Service-Oriented Architecture (SOA).
4. To categorize and assess security concerns within the domain of cloud computing.

Course Outcomes

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

1. Analyse the different cloud service models (IAAS, PAAS, SAAS) and deployment models for specific business needs
2. Apply virtualization concepts to create and manage virtual machines, demonstrating proficiency in virtualization technologies
3. Utilize cloud APIs to build practical solutions for cloud integration and management within a Service-Oriented Architecture (SOA) framework
4. Evaluate the core issues of Cloud Computing, such as Security, Privacy, Identity Management, and Access Control

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Cloud Services and Cloud Models** Introduction to Cloud, Cloud Computing vs. Cluster Computing vs. Grid Computing Introduction to Cloud Service Models: IAAS, PAAS, SAAS Characteristics, Benefits and Applications, Comparison of SAAS, PASS and IAAS XAAS- Anything as a Service – Storage as a service, Network as a Service, Database as a Service etc. Cloud Deployment Models-Public, Private, Hybrid Cloud Platforms: Google Cloud Platform, Microsoft Azure, SalesForce, AWS.	6

2	**Understanding Abstraction and Virtualization** Basics of Virtualization & Hypervisors Concept, Types of Virtualizations: Server, Storage and Network, Advantages and Disadvantages of Virtualization, Machine Image, Virtual Machine (VM), Open-Source Virtualization Technology Examples: VMware: Full Virtualization, Virtual Box	8
3	**Service Oriented Architecture and Cloud Management** Introducing Service-Oriented Architecture, Defining SOA Communications Cloud APIs (RESTful), Managing and Monitoring SOA Computing, Relating SOA and Cloud Computing	6
4	**Cloud Security** Securing the Cloud: The security boundary, Security service boundary, Security mapping Securing Data: Brokered cloud storage access, Storage location and tenancy, EncryptionEstablishing Identity and Presence	10
	Total	**30**

Text Books:

1. B. Sosinsky, Cloud Computing Bible, Wiley India, 2011

Reference Books:

1. B. Furht, “Handbook of Cloud Computing”, Springer,2010
2. A. T. Velte, T. J. Velte, R. Elsenpeter, Cloud Computing- A Practical Approach, McGrawHill Education, 2019.
3. T. Malhar, S. Kumaraswammy, S. Latif, Cloud Security & Privacy, SPD, O’REILLY, 2009
4. V. Josyula, ”Cloud computing – Automated virtualized data center”, CISCO Press, 2011
5. Dr. Kumar Saurabh , Cloud Computing, Wiley–India, 2015
6. M. Miller, Cloud Computing Web –Based Applications that change the way you work and Collaborate Online, Pearson, 2008
7. J. Hurwitz, R. Bloor, M. Kaufman, Cloud Computing for Dummies, FernHalper, 2011

E-Resources:

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

Program:	**MCA**						**Semester : II**	
Course:	**Blockchain Technology (PEC-2)**						**Code : MCA32PE11**	
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. Computer Network
2. Object Oriented Programming

is essential.

Course Objectives

This course aims at enabling students:

1. To give the basic overview of the Blockchain technology
2. To develop the skill and knowledge of Hyperledger

Course Outcomes

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

1. Explain the concepts related to Blockchain.
2. Understand various protocols involved in Blockchain enabled applications
3. Use Hyperledger to demonstrate the use of Blockchain
4. Discuss Blockchain use cases in various domains

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Introduction** History: Digital Money to Distributed Ledgers -Design Primitives: Protocols, Security, Consensus, Permissions, Privacy- : Block chain Architecture and Design-Basic crypto primitives: Hash, SignatureHash chain to Block chain-Basic consensus mechanisms.	7
2	**Consensus** Requirements for the consensus protocols-Proof of Work (PoW)-Scalability aspects of Block chain consensus protocols: Permissioned Block chains-Design goals-Consensus protocols for Permissioned Block chains.	7
3	**HyperLedger** Decomposing the consensus process-Hyper ledger fabric components-Chain code Design and Implementation: Hyper ledger Fabric II:-Beyond Chain code: fabric SDK and Front End-Hyper ledger composer tool.	8
4	**Blockchain Use Cases** Blockchain in Financial Service(Payments and Secure Trading, Compliance and Mortgage, Financial Trade), Blockchain in Supply Chain, Blockchain in Government (Advantages, Use Cases, Digital Identity), Blockchain Security	8

	(Overview, Membership and Access control in Fabric,Privacy in Fabric)	
	Total	**30**

Text Books:

1. Antonopoulos A., *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, O’Reilly Media, 2014
2. Gates M., *Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of Money*, Wise Fox Publishing, 2017

Reference Books:

1. Swan M., *Blockchain,* O’Reilly Media, 2014

E-Resources:

1. An IBM Redbook Course on Blockchain : https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html
2. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

Program:	**MCA**	**Semester : II**
Course:	**Business Opportunity Identification (PEC-2)**	**Code : MCA32PE12**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**FA**		**TW**	**SA**	**Total**
				FA-1	**FA-2**			
2	**2**	**-**	**-**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. Start Up and New Venture Development

is essential.

Course Objectives

This course aims at enabling students:

1. To inculcate an entrepreneurial mindset into the minds of young professionals
2. To identify entrepreneurial opportunities
3. To leverage skills for founding, leading & managing Startups
4. To understand the process of opportunity recognition and evaluation.
5. To learn the processes and practices in business and their applications

Course Outcomes

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

1. Apply market analysis frameworks to assess industry trends
2. Discover the different Business Ideas
3. Utilize techniques to identify different business opportunities
4. Analyze the Business Model Canvas to develop new business models

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Market Analysis and Trends** Understanding market dynamics,Analyzing industry trends and emerging markets,Identifying niche markets and underserved customer segments,Concept of Market Segmentation,Social responsibility and sustainability considerations	7
2	**Business Opportunity Identification** Concept of Business Opportunity,Business idea,Business Opportunities Identification Process,Business Value Chain, different sections of the business value chain for potential opportunities	7
3	**Business Opportunity Identification Techniques** Business Opportunities in India,Different Business Models,Identifying the right Business Model, Opportunities in different industries / Sectors Feasibility Analysis	8

4	**Design Thinking** Overview of Design Thinking, Key principles of Design Thinking, Applying Design Thinking in different contexts (business, social impact, education) ,Examination of successful Design Thinking case studies	8
	Total	**30**

Text Books:

1. Desai, V. *Dynamics of Entrepreneurship Development.* Himalaya Publishing House, (2017).
2. Holt, D. H. *Entrepreneurship: New Venture Creation.* Prentice Hall, (2019).

Reference Books:

1.Taneja, S., Gupta, S.L. *Entrepreneurship Development New Venture Creation*. S. Chand & Company Ltd, (2015).

2.Charantimath, P. M. *Entrepreneurship Development and Small Business Management.* Pearson Education India, (2016).

E-Resources:

1. https://www.startupindia.gov.in/
2. https://www.aim.gov.in/
3. https://www.investindia.gov.in/
4. https://msme.gov.in/
5. https://www.sidbi.in/

Program:	**MCA**						**Semester : II**
Course:	**Data Warehouse and Data Mining Lab (PEC-1 Lab)**						**Code : MCA32PE05**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. SQL queries and Designing ERD.
2. Data Mining Algorithms: Association Mining, Clustering and classification

is essential.

Course Objectives

This course aims at enabling students:

1. To understand fundamental concepts, techniques and design principles of data warehousing and data mining.
2. To enable students to understand, implement and evaluate various algorithms in data mining

Course Outcomes

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

1. Apply different preprocessing methods to prepare data in the desired format.
2. Create Data Warehouse schemas and apply OLAP operations for effective multidimensional data analysis.
3. Evaluate frequent patterns using Association Mining Techniques on large datasets.
4. Evaluate various Clustering and Classification Techniques on large datasets.

Guidelines: Students will be assessed based on

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

Students are advised to use:

open source/ freeware tool (Oracle Express Edition/ Oracle Live SQL) and Weka

Detailed Syllabus

Asgn. No.	**Suggested List of Assignments**
1	**Working with WEKA and Tanagra** 1. Install Tools: ○ Install both WEKA and Tanagra data mining tools on your computer. 2. Prepare Dataset: ○ Create a dataset in CSV format. ○ Convert the CSV file to ARFF format for use in WEKA. 3. Load and Explore in WEKA: ○ Import the dataset into WEKA. ○ Explore the dataset structure and attribute statistics. ○ Visualize class distributions to understand the data better. 4. Apply Preprocessing Techniques:

	○ Apply normalization to scale the data. ○ Apply discretization to convert numeric attributes into categorical values. ○ Observe and report the effects of these techniques on the dataset. 5. Compare with Tanagra: ○ Load the same dataset into Tanagra. ○ Perform similar analysis and compare the results with those from WEKA.
2	**Dimensional Modeling: Data Cube and OLAP Operations Assignment** **Design a data cube using either the Star Schema or Snowflake Schema model. Use the following dimension tables**: ● Product ● Customer ● Location Then, do the following tasks: 1. Write 20 analytical queries based on the data cube. 2. Perform the following **OLAP operations**: ○ Roll-Up ○ Roll-Down ○ Slice ○ Dice ○ Pivot Document your work and explain how each operation affects the view of the data.
3	**Association Rules Analysis** Use the supermarket dataset and apply the Apriori algorithm using its default settings. Examine the association rules that are generated and summarize your observations. Next, modify the algorithm settings to: ● Minimum Support = 0.3 ● Confidence = 0.8 Run the algorithm again with these new settings. Compare the new results with those from the default run. Write a brief report highlighting your main findings and include some of the most interesting and useful association rules identified during the experiment
4	**Classification:** Use the weather dataset and apply the Naive Bayes classification algorithm with the default settings. Review the classification rules generated and describe your observations. Next, evaluate the performance of the algorithm using the following testing methods: ● Training set ● Supplied test set ● Cross-validation ● Percentage split Write a brief report summarizing your findings and comparing the results from each evaluation method.

5	**Clustering:** Use the Iris dataset and apply the Simple K-Means algorithm &HierarchicalClusterer with its default settings. Observe and describe the clusters (groups) that are formed. Next, change the number of clusters to 4 and use the **Manhattan distance** instead of the default distance measure. Try the algorithms again with these new settings. Write a short report explaining what you discovered from both experiments. Compare the results and highlight any important differences or interesting patterns you noticed.

References:

1. J. Han, M. Kamber ,Data Mining: Concepts and Techniques.Morgan Kaufmann. (2011)
2. G.K.Gupta : Introduction to Data Mining with Case Studies.PHI.(2014)
3. Parteek Bhatia: Data Mining and Data Warehousing-Principles and Practical Techniques.Cambridge University Press.(2019)

Program:	**MCA**						**Semester : II**
Course:	**Web Development using Django Lab (PEC-1 Lab)**						**Code : MCA32PE06**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Basic Programming Concepts
2. Basics of Web Development
3. Logical Thinking
4. Database Operations

is essential.

Course Objectives

This course aims at enabling students:

1. Understand the concept of dynamic web pages and the Django template system.
2. Learn how to map URLs to views and handle 404 errors in Django.
3. Gain knowledge of the Model-View-Template (MVT) development pattern in Django.
4. Develop skills in defining and installing database models using Django's ORM.
5. Explore the Django admin interface and its customization capabilities.
6. Learn about form processing, validation, and saving form responses in Django.
7. Understand access control mechanisms using sessions and user authentication in Django.
8. Implement login, logout, user management, and permission handling in Django applications.
9. Gain exposure to asynchronous messaging concepts.

Course Outcomes

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

1. Create dynamic web pages using Django's template system and context objects.
2. Implement database models.
3. Implement Django admin interface for efficient management of application data, including users, groups, permissions, and form processing.
4. Implement the Django session framework to manage user sessions, handle session data in views, optimize session usage for performance, and ensure secure session management practices.

Guidelines:

Students will be assessed based on

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

Students are advised to use:

A. PyCharm
B. Python 3.X

Detailed Syllabus

Assign. No.	Suggested List of Assignments
1	Set up a Django project and app, Create a basic template with static content, Render the template with dynamic data using context objects, Implement URL mapping and handle 404 errors
2	Define a model with fields and options, Install and migrate the model to the database, Create model instances and perform CRUD operations, Customize model string representations
3	Activate the Django admin interface, Register models with the admin interface, Perform CRUD operations using the admin interface, Customize the admin interface's look and feel
4	Create a form model and template, Implement form validation and error handling, Process form data and save to the database
5	Set up user authentication in Django, Implement login and logout views, Manage users, groups, and permissions, Restrict access to views based on user roles
6	Set up a message queue (e.g., RabbitMQ or Redis), Implement asynchronous task processing using a task queue, Monitor and manage asynchronous tasks

References:

1. Greenfeld, D. R., &Greenfeld, A. R. *Two Scoops of Django 1.11: Best Practices for the Django Web Framework.* Two Scoops Press, (2017)
2. Vincent, W. S. *Django for Beginners: Build Websites with Python and Django. WelcomeToCode*, (2021)
3. **Django Tutorial :** https://www.w3schools.com/django/
4. **Learn Django Framework :** https://www.geeksforgeeks.org/django-tutorial/
5. **Django Tutorial :** https://www.tutorialspoint.com/django/index.htm

Program:	**MCA**	**Semester : II**
Course:	**UI/UX Design Lab (PEC-1 Lab)**	**Code : MCA32PE07**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. HTML
2. CSS
3. Basics of Design Principles

is essential.

Course Objectives

This course aims at enabling students:

1. Provide hands-on experience in applying UI/UX design principles and techniques to real-world design projects.
2. Foster creativity and critical thinking skills in solving design problems and challenges.
3. Develop proficiency in using industry-standard design tools and software.
4. Cultivate collaboration and communication skills through teamwork and peer feedback.
5. Prepare students for careers in UI/UX design through practical project experience and portfolio development.

Course Outcomes

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

1. Apply user-centered design principles to create intuitive and user-friendly interfaces.
2. Demonstrate proficiency in using design tools such as Adobe XD, Sketch, or Figma to create wireframes, prototypes, and mockups.
3. Conduct user research and usability testing to inform design decisions and iterate on designs based on feedback.
4. Create a professional portfolio showcasing their UI/UX design skills, process, and completed projects.

Guidelines:

Students will be assessed based on

A. the practical work done by them throughout the semester
B. the Practical Exam
C. the Oral Exam.

Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Detailed Syllabus

Asgn. No.	**Suggested List of Assignments**
1	Introduction to UI/UX Design : Overview of UI/UX design principles and process, Introduction to design tools (e.g., Adobe XD, Sketch, Figma), Project kickoff and team formation

2	Design Project Sprint 1 : Conducting user research: interviews, surveys, and usability testing, Creating user personas and journey maps, Ideation and sketching exercises, Wireframing and prototyping
3	Design Project Sprint 2 : Refining wireframes and prototypes based on user feedback, Visual design: typography, color, and branding, Creating high-fidelity mockups and interactive prototypes, Usability testing and iteration
4	Design Project Sprint 3 : Finalizing designs and preparing for presentation, Peer feedback sessions and design critiques, Presenting final designs to class and stakeholders, Portfolio development: documenting design process and completed projects
5	Portfolio Development and Review : Building a professional portfolio website, Crafting case studies for design projects, Peer review and feedback on portfolio presentation
6	Final Project Showcase : Final presentation of portfolios and design projects, Reflection on learning outcomes and course experience

References:

1. Krug S., Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, New Riders Publication, 2014
2. Norman D., The Design of Everyday Things, Basic Books Publication, 2013
3. Garrett J. J., The Elements of User Experience: User-Centered Design for the Web and Beyond, New Riders Publication, 2010
4. Knight W., UX for Developers, Apress Publication, 2018
5. UI/UX and Figma : https://www.freecodecamp.org/news/ui-ux-design-tutorial-from-zero-to-hero-with-wireframe-prototype-figma/
6. UI/UX Tutorial : https://intellipaat.com/blog/ui-ux-design-tutorial

Program:	**MCA**	**Semester : II**
Course:	**Start Up and New Venture Development Lab (PEC-1 Lab)**	**Code : MCA32PE08**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

- No Prior knowledge is required

Course Objectives

This course aims at enabling students:

1. To develop entrepreneurship awareness
2. To inculcate an entrepreneurial mindset into the minds of young professionals
3. To identify entrepreneurial opportunities
4. To leverage skills for founding, leading & managing Startups

Course Outcomes

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

1. Develop traits and factors influencing the development of entrepreneurship as a profession
2. Develop a strategic plan for launching a Startup
3. Discover skill sets required for successful Entrepreneurship
4. Summarize the role of Government's various support organizations in promoting Entrepreneurship

Guidelines:

Students will be assessed based on

A. The participation in all activities by them throughout the semester,
B. The Evaluation Exam, and
C. The Oral Exam.

The participation in all activities and the Evaluation Exam collectively has the weightage of 30 marks and oral exam has the weightage of 20 marks.

Detailed Syllabus

Asgn. No.	**Suggested List of Assignments**
1	Case Study of Successful Entrepreneurs
2	Case Study of Failed Businesses
3	Study of First-Generation Entrepreneurs
4	Skills for Successful Entrepreneurs

5	Case Study on Start Up India

References:

1. Desai, V. *Dynamics of Entrepreneurship Development.* Himalaya Publishing House, (2017).
2. Holt, D. H. *Entrepreneurship: New Venture Creation.* Prentice Hall, (2019).
3. Taneja, S., Gupta, S.L. *Entrepreneurship Development New Venture Creation*. S. Chand & Company Ltd, (2015).
4. Charantimath, P. M. *Entrepreneurship Development and Small Business Management*. Pearson Education India, (2016).
5. https://www.startupindia.gov.in/
6. https://www.aim.gov.in/
7. https://msme.gov.in/
8. https://www.sidbi.in/

Program:	MCA						Semester : II
Course:	**Data Science Lab (PEC-2 Lab)**						**Code : MCA32PE13**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Basics of Python Programming
2. Basics of Business Statistics

is essential.

Course Objectives

This course aims at enabling students:

1. To develop relevant Python programming abilities for analysis of data.
2. To learn and demonstrate the basic statistical analysis of data.
3. To understand the features and applications of machine learning algorithms.
4. To use appropriate tools and methods to collect, process, summarize, and visualize data for analysis

Course Outcomes

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

1. Understand the basic concepts of the Data Science
2. Apply basic statistical operations using Python programming libraries on datasets.
3. Analyze and implement data visualization techniques for data analysis.
4. Analyze and implement machine learning techniques for data analysis and visualization.

Guidelines:

1. Students will be assessed based on:
   A. The practical work done by them throughout the semester
   B. The Practical Exam, and
   C. The Oral Exam.

Practical work and practical exams collectively have a weightage of 30 marks, and oral exam has a weightage of 20 marks.

Students are advised to use:

a) Version control system such as Git
b) Google Colab. for Cloud-based and browser-based Python Interpreter
c) Use of VS Code editor is recommended. Students may also use alternatives such as Python IDLE, PyCharm, etc

Detailed Syllabus

Asgn. No.	Suggested List of Assignments
1	Demonstration of Python programming and Python Interpreter
2	Assignments based on Data Importing and Exporting for Data Analysis

3	Assignment based on business statistics python library and central tendencies
4	Assignment based on probability, distribution, variance, correlation, and standard deviation
5	Demonstration of NumPy and Pandas Libraries - Assignment based on Numpy Library
6	Demonstration of NumPy and Pandas Libraries - Assignment based on Numpy Library
7	Demonstration of data visualization using MatPlotLib and Seaborn libraries - Assignment based on data visualization
8	Assignment based on data preprocessing, training and testing for analysis
9	Demonstration of Machine Learning Algorithms (Supervised & Unsupervised) - Assignment based on Machine Learning Concepts for data preprocessing
10	Demonstration of Regression and Classification Algorithms - Assignment based on Machine Learning Algorithms

References:

1. Fabio Nelli, *Python for Data Analytics with Pandas, Numpy, and MatPlotLib*, Apress Publication, 2023.
2. Wes McKinney, *Python for Data Analysis*, O'Reilly Publication, 2017.
3. Naval Bajpai, *Business Statistics*, Pearson Publication. 2010.
4. Google's online Jupyter Notebook, https://colab.research.google.com/
5. Python Tutorial: https://docs.python.org/3/tutorial/index.html

Program:	**MCA**	**Semester : II**
Course:	**Cloud Computing Lab (PEC-2 Lab)**	**Code : MCA32PE14**

Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

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

is essential.

Course Objectives

This course aims at enabling students:

1. To provide an overview of cloud computing, encompassing its core principles, technologies, challenges, and real-world applications.
2. To understand the virtualization technologies and architectural concepts.
3. To explore the interconnectedness between cloud computing and Service-Oriented Architecture (SOA).
4. To categorize and assess security concerns within the domain of cloud computing

Course Outcomes

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

1. Analyse the different cloud service models (IAAS, PAAS, SAAS) and deployment models for specific business needs.
2. Apply virtualization concepts to create and manage virtual machines, demonstrating proficiency in virtualization technologies.
3. Utilize cloud APIs to build practical solutions for cloud integration and management within a Service-Oriented Architecture (SOA) framework.
4. Evaluate the core issues of Cloud Computing, such as Security, Privacy, Identity Management, and Access Control.

Guidelines:

Students will be assessed based on

A. The practical work done by them throughout the semester,
B. The Practical Exam, and
C. The Oral Exam.

Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Detailed Syllabus

Asgn. No.	**Suggested List of Assignments**
1	**Cloud Services and Cloud Models** Cloud Platform Exploration Create accounts on: Google Cloud, Microsoft Azure, Salesforce, AWS.

2	**XaaS Implementation:** Develop a prototype of a specific "as-a-Service" (XaaS) model, such as Storage as a Service or Database as a Service.
3	**Setting Up a Virtualization Environment:** Install a hypervisor (e.g., VirtualBox) and create virtual machines.
4	Explore the management and configuration of VMs
5	Design a simple SOA-based application.
6	Develop and deploy web services using RESTful APIs
7	Configure virtual networks in a cloud environment.
8	Implement firewall rules and network security settings
9	Monitor resource usage on a cloud platform.
10	Moving Applications to the Cloud.

References:

1. B. Sosinsky, Cloud Computing Bible, Wiley India, 2011
2. B. Furht, “Handbook of Cloud Computing”, Springer, 2010
3. A. T. Velte, T. J. Velte, R. Elsenpeter, Cloud Computing- A Practical Approach, McGrawHill Education. 2019
4. T. Malhar, S. Kumaraswammy, S. Latif, Cloud Security & Privacy, SPD, O’REILLY, 2009
5. V. Josyula, ”Cloud computing – Automated virtualized data center”, CISCO Press, 2011
6. Dr. Kumar Saurabh , Cloud Computing, Wiley–India, 2015
7. M. Miller, Cloud Computing Web –Based Applications that change the way you work and Collaborate Online, Pearson, 2008
8. J. Hurwitz, R. Bloor, M. Kaufman, Cloud Computing for Dummies, FernHalper, 2011
9. Cloud Computing Tutorial https://www.tutorialspoint.com/cloud_computing/cloud_computing_tutorial.pdf

Program:	**MCA**						**Semester : II**
Course:	**Blockchain Technology Lab (PEC-2 Lab)**						**Code : MCA32PE15**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Computer Networks
2. Object Oriented Programming

is essential.

Course Objectives

This course aims at enabling students:

1. To make the students aware about HyperLedger, Smart Contracts, Etherium
2. To expose the students to various applications of Blockchain

Course Outcomes

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

1. Implement basics of Blockchain
2. Simulate the use of Blockchain in different domains
3. Compile and deploy smart contracts
4. Setup the network using HyperLedger
5. DevelopDecentralized Finance (DeFi) Smart Contract

Guidelines:

Students will be assessed based on

A. The practical work done by them throughout the semester
B. The Practical Exam
C. The Oral Exam.

Practical work and practical exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks.

Detailed Syllabus

Assign. No.	**Suggested List of Assignments**
1	Exercises based on Blockchain Basics
2	Exercises based on Smart Contract
3	Exercises based on Ethereum
4	Exercises based on HyperLedger
5	Exercises based on Decentralized Finance (DeFi) Smart Contract

References:

1. Antonopoulos A., Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O’Reilly Media, 014
2. Gates M., Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of Money, Wise Fox Publishing, 2017
3. Swan M., Blockchain, O’Reilly Media, 2014
4. An IBM Redbook Course on Blockchain : https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html
5. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

Program:	**MCA**						**Semester : II**
Course:	**Business Opportunity Identification Lab (PEC-2 Lab)**						**Code : MCA32PE16**
Credits	**Teaching Scheme (Hrs. / Week)**			**Evaluation Scheme and Marks**			
	Theory	**Practical**	**Tutorial / Activity**	**TW**	**OR**	**PR**	**Total**
2	**-**	**4**	**-**	**-**	**20**	**30**	**50**

Prior knowledge of :

1. Start Up and New Venture Development

is essential.

Course Objectives

This course aims at enabling students:

1. To inculcate an entrepreneurial mindset into the minds of young professionals
2. To identify entrepreneurial opportunities
3. To leverage skills for founding, leading & managing Startups
4. To understand the process of opportunity recognition and evaluation.
5. To learn the processes and practices in business and their applications

Course Outcomes

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

1. Apply market analysis frameworks to assess industry trends
2. Discover the different Business Ideas
3. Utilize techniques to identify different business opportunities
4. Analyze the Business Model Canvas to develop new business models

Guidelines:

Students will be assessed based on

A. The participation in all activities by them throughout the semester,
B. The Evaluation Exam, and
C. The Oral Exam.

The participation in all activities and the Evaluation Exam collectively have the weightage of 30 marks and oral exam has the weightage of 20 marks**.**

Detailed Syllabus

Assign. No.	**Suggested List of Assignments**
1	Activity based on Market Analysis and Market Segmentation
2	Exercises based on Business Opportunity Identification
3	Exercises based on Identifying the Right Business Model

4	Exercises based on Business Model Canvas Framework
5	Exercises based on Design Thinking

References:

1. Desai, V. Dynamics of Entrepreneurship Development. Himalaya Publishing House, 2017.
2. Holt, D. H. Entrepreneurship: New Venture Creation. Prentice Hall, 2019.
3. Taneja, S., Gupta, S.L. Entrepreneurship Development New Venture Creation. S. Chand & Company Ltd, 2015.
4. Charantimath, P. M. Entrepreneurship Development and Small Business Management. Pearson Education India, 2016.
5. https://www.startupindia.gov.in/
6. https://www.aim.gov.in/
7. https://msme.gov.in/
8. https://www.sidbi.in/

Program:	**MCA**	**Semester : II**
Course:	**Project Management**	**Code : MCA32EM01**

Credits	**Teaching Scheme (Hrs. / Week)**				**Evaluation Scheme and Marks**				
					FA				
	Lecture	**Practical**	**Tutorial / Activity**	**O***	**FA-1**	**FA-2**	**TW**	**SA**	**Total**
2	**2**	**-**	**-**	**4**	**10**	**10**	**-**	**30**	**50**

Prior knowledge of :

1. Problem-solving and Analytical Thinking
2. Mathematics
3. Understanding of Basic Management Concepts
4. Concepts of Software Engineering and Software Testing

is essential.

Course Objectives

This course aims at enabling students:

1. To understand the fundamentals of Software Project Management
2. To investigate software project planning and agile project management
3. To learn Risk Management and Problem-Solving
4. To Enhance Leadership and Team Management Skills

Course Outcomes

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

1. Comprehend Project Management Concepts & Agile Project Management
2. Choose various techniques of Software Project Estimations
3. Gain knowledge of configuration management
4. Analyze staffing process for team

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Introduction to Software Project Management** Overview of project Management,Project management life cycle,Agile Project Management, Risk Management Risk Management Process	6
2	**Project Planning & cost Estimation** Project planning,Gantt Chart, CPM,PERT Chart,Different methods of Cost estimation,COCOMOmodel,Function Point Analysis,Delphi cost estimation ,Software Project Metrics	10
3	**Configuration Management** Configuration management & Maintenance plan , Change Management, Version and Release Management Configuration Management Tools Configuration Management in Agile	6
4	**Staffing in Software Projects** Organizational structures Team Structure & Staff development plan ,	8

	Characteristics of Performance management ,High performance Directive and collaborative styles , Team Communication ,Group Behaviors , Managing customer expectations ,Agile Team Management, Emerging Trends in Staffing	
Other*	Microsoft Project,Jira support agile methodologies,Branching and release tagging in projects	4
	Total	**30+4=34**

Text Books:

1. Hughes, B., Cotterell, M., & Mall, R. Software Project Management (6th ed.). Tata McGraw Hill, 2017.
2. Wysocki, R. K. Effective Software Project Management. Wiley, 2011.

Reference Books:

1. Pressman, R. S. Software Engineering: A Practitioner's Approach. McGraw Hill, 2014.
2. Schwaber, K. Agile Project Management with Scrum. Microsoft Press, 2004.
3. Royce, W. Software Project Management: A Unified Framework. Addison-Wesley, 1998.
4. Jalote, P. Software Project Management in Practice. Addison-Wesley Professional, 2002.
5. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). Project Management Institute, 2017.

E-Resources:

1. https://www.computer.org/education/bodies-of-knowledge/software-engineering
2. https://www.pmi.org/pmbok-guide-standards/foundational/pmbok
3. https://www.mindtools.com/ct91phy/project-management
4. https://www.atlassian.com/agile/project-management

Program:	**MCA**	**Semester : II**
Course:	**Mathematical Foundation for Computer Application-2**	**Code : MCA32BS02**

Credits	**Teaching Scheme (Hrs. / Week)**				**Evaluation Scheme and Marks**				
	Lecture	**Practical**	**Tutorial / Activity**	**O***	**FA**		**TW**	**SA**	**Total**
					FA-1	**FA-2**			
4	**3**	**-**	**1**	**6**	**20**	**20**	**-**	**60**	**100**

Prior knowledge of :

1. Basic Mathematics
2. Probability

is essential.

Course Objectives

This course aims at enabling students:

1. To learn the basics of business decision-analysis.
2. To summarize business data numerically and graphically.
3. To understand the importance of business sampling methods, and be able to describe different business sampling methods.
4. To understand the process associated with statistical decisions, defining and formulating problems, analyzing the data, and using the results in decision making.

Course Outcomes

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

1. Apply the concepts of statistics, data representation and Measures of Central Tendency.
2. Determine Sampling and Sampling Distribution
3. Implement various hypothesis testing techniques.
4. Illustrate correlation between the attributes
5. Articulate concepts of time series moving average.

Detailed Syllabus

Unit No.	**Description**	**Duration (Hrs.)**
1	**Introduction to statistics** Importance of statistics in modern business environment ,Definition of statistics, importance, scope and applications ,Characteristics of statistics, Functions of Statistics, Limitations ,Need of Data, Types of Data Principles of Measurement,Source of Data ,Data classification, Tabulation And presentation	7
2	**Measures Of Central Tendency** Introduction, Objectives of Statistical average Requisites of a Good Average ,Statistical Averages- Arithmetic Mean ,Properties Of A M, Mean of combined group, Median, Mode, Geometric mean and Harmonic Mean.	7
3	**Sampling, Sampling Distributions And Testing** Introduction ,Population And Sample-Universe of Population ,Types Of Population- Sample, Advantages of Sampling ,Sampling Theory- Types Of Sampling.	6

4	**Testing Of Hypothesis** Introduction Testing Hypothesis ,Classification Of Test statistics Testing of Hypothesis ,Z-Test, ‘t’ test ,chi square proportion test	9
5	**Simple Correlation And Regression** Introduction Correlation-Types of Correlation-measures of correlation-Properties Of Karl Pearson’s correlation coefficient Spearman’s Rank Correlation coefficient Regression- Regression analysis	8
6	**Time Series Analysis** Introduction Utility of the time series ,Components of Time Series Methods of measuring trend ,Method of least squares ,Mathematical Models of Time series ,Forecasting methods using time series	8
Other*	Applicability of Data Analysis on various dataset to solve real life problem, Understanding of ANOVA, F-test	6
	Total	**45+6=51**

List of Tutorial / Activities

1. Activity on Importance of statistics
2. Activity on Measures Of Central Tendency
3. Activity on Sample Distribution
4. Activity on Testing Hypothesis
5. Activity on Simple Correlation and Regression
6. Activity on Time Series Analysis

Text Books:

1. Gupta, S. C., &Kapoor, V. K..*Fundamentals of mathematical statistics*. Sultan Chand & Sons.2020.
2. Williams, A. S. W. A. S. Statistics for business and economics. *A TRIZ*.2011.

Reference Books:

1. Gupta, S. P. . Statistical Methods By SP Gupta. *Sultan Chand and Sons, Delhi, India*.2001.
2. Gupta, N. *Business Statistics*, Naval Bajpai, Pearson Education. 2009.
3. Aczel, A. D., &Sounderpandian, J..*Complete business statistics*. McGraw Hill. 1999.
4. Jani, P. N. *Business statistics: Theory and applications*. PHI Learning Pvt. Ltd.2014.

E-Resources:

1. Statistical Concept:Statistical Methods calculators (atozmath.com)
2. Discrete Mathematics: https://www.tutorialspoint.com/discrete_mathematics/index.htm

Program:	MCA						Semester : II
Course:	Research Project						Code : MCA32EL02
Credits	Teaching Scheme (Hrs. / Week)			Evaluation Scheme and Marks			
	Theory	Practical	Tutorial / Activity	TW	OR	PR	Total
2	-	4	-	50	-	-	50

Course Objectives

This course aims at enabling students:

1. To introduce students a software engineering approach thorough problem analysis.
2. To gain an in-depth understanding of specific functional areas.
3. To enhance the ability of technical writing

Course Outcomes

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

1. Formulate the requirements based on problem statements.
2. Apply knowledge of Research Methodology for research analysis and/or design
3. Present research findings, both in written and verbal formats

About Course:

Research Project provides an opportunity to students to develop practical skills and systematic work undertaken in the fields of computer application. This course focuses on practical work to develop problem-solving skills in computing. It helps students develop research and technical writing skills.

Guidelines (Research Work): For projects involving Research Work with a 50-mark evaluation, the following guidelines are to be followed:

1. Students undertaking research projects are encouraged to choose topics relevant to their field of study.
2. The Research Project Synopsis should contain an Introduction to Project, Objectives, and Problem statement, which should clearly explain the project scope in detail. Any other point may be included if required as per the project type/category/nature/scope.
3. Students are required to present the progress of the Research Project during the semester as per the project review schedule.
4. The evaluation of the Research Project will be based on minimum 2 (two) reviews and will carry a total of 50 marks.
5. Research work may involve carrying out a substantial research-based project or activity, analyzing data, research findings and explaining research articles.
6. At the end of the semester, students must submit the Research project report as per the template.

VISION AND MISSION OF MCA DEPARTMENT

VISION:

To be a renowned hub in Computer Application education dedicated to provide Ethical, Sustainable, and Value-Added Education, nurturing skilled professionals equipped for successful careers in the digital world.

MISSION:

- Cultivate a responsive community that addresses the evolving needs of society by providing essential learning environment and culture
- Empower students with the right Attitude, Skills, and Knowledge by providing a comprehensive curriculum in computer applications development and management
- Foster the growth of globally competent and thoughtful graduates, researchers, and entrepreneurs who excel in demanding scenarios