NOIDA INSTITUTE OF ENGGNEERING & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR
(AN AUTONOMOUS INSTITUTE)

NIET
Greater Noida
Estd.2001
Get Future Ready
Autonomous Institute

Affiliated to

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

Evaluation Scheme & Syllabus

For

Master of Computer Applications (Integrated)
Second Year

(Effective from the Session: 2025-26)

NOIDA INSTITUTE OF ENGGNEERING & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR
(AN AUTONOMOUS INSTITUTE)
Master of Computer Applications
MCA-INT
Evaluation Scheme
SEMESTER III

S. No	Subject Codes	Subjects	Types of Subjects	Periods			Evaluation Schemes				End Semester		Total	Credit
				L	T	P	CT	TA	Total	PS	TE	PE		
1	BMICA0304	Operating System	Mandatory	3	1	0	30	20	50		100		150	4
2	BMICA0302	Data Structures	Mandatory	3	1	0	30	20	50		100		150	4
3	BMICA0301	Accounting and Financial Management	Mandatory	3	1	0	30	20	50		100		150	4
4	BMICA0303	Emerging Trends and Technology	Mandatory	4	0	0	30	20	50		100		150	4
5	BMICA0355	Advance Python	Mandatory	0	0	8				50		100	150	4
6	BMICA0354	Operating System Lab	Mandatory	0	0	4				50		50	100	2
7	BMICA0352	Data Structures Lab	Mandatory	0	0	4				50		50	100	2
8	BMICA0359	Internship Assessment-I	Mandatory	0	0	2				50		50	100	1
		*Massive Open Online Courses	*MOOCs											
		TOTAL											**1050**	**25**

PLEASE NOTE: -

- **A 2-3 weeks Internship shall be conducted during summer break after semester-II and will be assessed during semester-III**

*** List of MOOCs Based Recommended Courses for Second year (Semester-III) MCA(Int) Students**

S. No.	Subject Code	Course Name	University / Industry Partner Name	No of Hours	Credits
1	BMC0012	Data Structures and Algorithms using Python - Part 1	Infosys Wingspan (Infosys Springboard)	29h 27m	
2	BMC0040	Data Structures and Algorithms using Python - Part 2	Infosys Wingspan (Infosys Springboard)	37h 41m	

Abbreviation Used:

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam., CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit, MOOCs: Massive Open Online Courses.

.

NOIDA INSTITUTE OF ENGGNEERING & TECHNOLOGY, GREATER NOIDA, GAUTAM BUDDH NAGAR
(AN AUTONOMOUS INSTITUTE)
Master of Computer Applications
MCA-INT
Evaluation Scheme
SEMESTER IV

S. No	Subject Codes	Subjects	Types of Subjects	Periods			Evaluation Schemes				End Semester		Total	Credit
				L	T	P	CT	TA	Total	PS	TE	PE		
1	BMICA0404	Design Thinking-II	Mandatory	2	1	0	30	20	50		100		150	3
2	BMICA0401	Cloud Computing	Mandatory	3	1	0	30	20	50		100		150	4
3	BMICA0402	Database Management System	Mandatory	3	1	0	30	20	50		100		150	4
4	BMICA0403	Software Engineering	Mandatory	4	0	0	30	20	50		100		150	4
5	BMICA0455	Object Oriented Techniques using Java	Mandatory	0	0	8				50		100	150	4
6	BMICA0452	Database Management System Lab	Mandatory	0	0	4				50		50	100	2
7	BMICA0451	Cloud Computing Lab	Mandatory	0	0	4				50		50	100	2
8	BMICA0459	Project Based on Software Engineering Lab	Mandatory	0	0	4				50		50	100	2
		*Massive Open Online Courses	*MOOCs											
		TOTAL											**1050**	**25**

*** List of MOOCs Based Recommended Courses for Second year (Semester-IV) MCA(Int) Students**

S. No.	Subject Code	Course Name	University / Industry Partner Name	No. of Hours	Credits
1	BMC0026	Introduction to Cloud Computing	Infosys Wingspan (Infosys Springboard)	5h 44m	
2	BMC0030	Software Engineering	Infosys Wingspan (Infosys Springboard)	8h 30m	

Abbreviation Used:

L: Lecture, T: Tutorial, P: Practical, CT: Class Test, TA: Teacher Assessment, PS: Practical Sessional, TE: Theory End Semester Exam., CE: Core Elective, OE: Open Elective, DE: Departmental Elective, PE: Practical End Semester Exam, CA: Compulsory Audit, MOOCs: Massive Open Online Courses.

.

NIET Greater Noida Autonomous Institute	**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY** **GREATER NOIDA-201306** **(An Autonomous Institute)** **School of Computer Applications**

Subject Name: Operating Systems	**L-T-P [3-1-0]**
Subject Code: BMICA0304	**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: Students are expected to be familiar with computer organization.	
Course Objective: The objective of this course is to provide an understanding of the basic structure and functions of an operating system and deliver the skills needed to develop Unix/Linux shell programs.	

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Describe operating system concepts, functions and design CPU Scheduling algorithms	**K2, K5**
CO2	Analyse the various issues related to inter process communication like Synchronization and Deadlocks.	**K4**
CO3	Describe the concepts of Memory Management and Implement disk scheduling algorithms.	**K5**
CO4	Design and use Linux utilities to create and manage simple file processing operations.	**K3, K4**
CO5	Implement shell scripts to perform more complex tasks in shell programming environment.	**K3**

Syllabus

Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**
UNIT-1	**Fundamentals of Operating Systems**	Operating System, Operatic System characteristics, Functions of Operating Systems, Types of Operating System, Layered Structure, System call, Kernel, Multiprogramming and		8L+4P	Assignment, Lab	**CO1**

Fundamentals of Operating Systems		Multitasking, Overview of Windows OS, Unix/Linux OS Process Management	Lectures, PPTs, Notes		(26 -29)	
	Process Management	Process Concepts, State Transition Diagram. Types of Schedulers: Long Term, Mid Term, Short Term Process Control Block, Inter process communication Process Management: Process Concepts, State Transition Diagram.				
	CPU Scheduling	CPU Scheduling Criteria, Pre-emptive and Non-Pre-emptive Scheduling, Scheduling Algorithm: FCFS, SJF, SRTF, Round Robin, Priority Scheduling, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling, Context Switching.				
UNIT-2 **Process Synchronization**	**Process Synchronization**	Critical Section problem & their solutions, Introduction to Semaphores Classical Problems of Synchronization (Producer Consumer Problem, Readers Writer Problem, Dining philosophers' problem)	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (30-33)	**CO2**
	Dead Locks	Characterization, Deadlock concepts & Handling Techniques (Prevention and Detection & Recovery), Dead Lock Avoidance: Banker's Algorithm.				
UNIT-3 **Memory management**	**Memory Management**	Memory Management: Background, Swapping, Contiguous and Non-Contiguous memory allocation, Paging, Segmentation, Segmentation with paging.	Lectures, PPTs, Notes	10L+4P	Assignment, Lab (34 to 45)	**CO3**
	Virtual Memory	Background, Demand paging,Allocation of frames: First Fit, Best Fit, and Worst Fit, Page replacement algorithms (FCFS, Optimal, LRU), Balady's Anomaly, Thrashing Disk Scheduling				
	Disk Scheduling	FCFS, SSTF, SCAN, C-SCAN, LOOK and C- LOOK				
	File Management System	File Management: Concept and Organization, Access Methods, File System Implementation Directory Structures, Allocation Methods, Free Space Management, Secondary Storage Structure, File System Security and Protection.				
UNIT-4	**Linux Administration**	Linux Components, Shells, Installation of Linux,	Lectures, PPTs, Notes	10L+4P	Assignment, Lab	**CO4**

Linux Administration	**Virtualization**	Definition, Types, Advantages, Virtualization tools. User Administration			(1-7)	
	Files&Commands	Files: Type, Ownership, Permissions and manipulations Commands: Internal and External, Directory and File commands, I/O commands, Pipes, Filters, shell commands				
	Linux Tools	Linux Networking Commands: ipconfig, traceroute, tracepath, ping, host, hostname, iwconfig. System Admin: man, uptime, users, service, pkill, ps.				
UNIT-5 **Shell Programming & vi Editor**	**Shell Programming vi Editor**	Shell Programming - shell script features, shell variables, writing and executing a shell script, positional parameters. Introduction to vi editor, vi editor Models, invoking vi editor, Configuring the vi environment, The process - parent and child process, process creation, process related commands, Branching control structures- if, case, Loop control structures– while, until, for, jumping control structures – break, continue, exit, Integer and Real arithmetic in shell programs	Lectures, PPTs, Notes	10L+4P	Assignment, Lab (8-25)	**CO5**
Total				**46L+20P**		

Textbooks	
Sr No	**Book Details**
1.	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts Essentials” 8th Edition, (2010)
2.	Andrew S. Tanenbaum, “Modern Operating Systems”, Pearson Education, 4th Edition, (2014)
3.	Jason Cannon, “Linux for Beginners: An Introduction to the Linux Operating System and Command Line”, (2014)
4.	Marks G. Sobell, “A practical guide to Linux: Commands, Editors and Shell Programming” Fourth Edition, (2017)
Reference Books	

Sr No	Book Details
1.	William Stallings ,“Operating Systems: Internals and Design Principles,”, 8th Edition, (2014)
2.	Charles Patrick Crowley ,“Operating System: A Design-oriented Approach”, 9th Edition, (2017)
3.	Gary J. Nutt, “Operating Systems: A Modern Perspective,” (1997)
4.	Maurice J. Bach., “Design of the UNIX Operating Systems,” 1st Edition, (2015)
5.	Daniel Pierre Bovet, Marco, “Understanding the Linux Kernel”1st Edition, (2000)
6.	AS Tanenbaum, AS Woodhull, Operating Systems Design and Implementation, 3rd Ed., Prentice Hall, (2006).
Links: NPTEL/ YouTube/ Faculty Video Link	
https://www.youtube.com/watch?v=783KAB-tuE4	
https://www.youtube.com/watch?v=ZaGGKFCLNc0	
https://nptel.ac.in/courses/106/105/106105214/	
https://www.youtube.com/watch?v=NShBeqTkXnQ	
https://www.youtube.com/watch?v=4hCih9eLc7M	
https://www.youtube.com/watch?v=9YRxhlvt9Zo	
https://www.youtube.com/watch?v=UczJ7misUEk	
https://www.youtube.com/watch?v=_IxqinTs2Yo	
https://www.youtube.com/watch?v=IwESijQs9sM	
https://www.youtube.com/watch?v=-orfFhvNBzY	

Subject Name: Data Structures						**L-T-P [3-1-0]**
Subject Code: BMICA0302						**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: Knowledge of programming languages, basics of mathematics, organizing and problem-solving ability.						
Course Objective: Learn the basic concepts of algorithm analysis, along with implementation of linear and non- linear data structures.						
Course Outcomes (CO)						
Course outcome: After completion of this course students will be able to:						**Bloom's Knowledge Level (KL)**
CO1	Describe the need of data structure and algorithms in problem solving and Analyse Time space trade-off.					**K3,K4**
CO2	Describe the real-world applications using stack and queue.					**K2**
CO3	Discuss different Linked list operations.					**K2**
CO4	Evaluate the real-world applications using non-linear data structures.					**K3,K4**
CO5	Identify and analyse the computational efficiencies of searching and sorting algorithms in real world problems					**K4,K5**
Syllabus						
Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**
UNIT 1 Introduction to Data Structures	**Data Types**	Types of Data Structures- Linear & Non-Linear Data Structures, List, Tuple, Set, Dictionary. Arrays: Derivation of Index Formulae for 1-D,2-D,3-D and n-D Array	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (1 to 10)	**CO1**
	Analysis of algorithms	Time and Space Complexity of an algorithm, Asymptotic notations (Big Oh, Big Theta and Big Omega).				
UNIT 2	**Stacks**	Primitive Stack operations: Push & Pop, mutual conversion of Infix, Prefix, Postfix, Evaluation of postfix expression		8L+4P	Assignment, Lab	**CO2**

Stack & Queues	**Recursion**	Principles of recursion, Types of Recursions, Problem solving using iteration, Tower of Hanoi,Trade-offs between iteration and recursion.	Lectures, PPTs, Notes		(11 to 20)	
	Queues	Operations on Queue: Create, Insert, Delete, Full and Empty, Circular queues, Dequeue and Priority Queue.				
UNIT 3 **Linked Lists**	**Linked lists**	Linked lists: Comparison of Array, List and Linked list Types of linked list: Singly Linked List, Doubly Linked List, Circular Linked List, Polynomial Representation and Addition of Polynomials**.**	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (21 to 30)	**CO3**
UNIT 4 **Trees**	**Trees**	Trees: Basic terminology, Binary Trees, Binary Tree Representation, Binary Search Tree, Strictly Binary Tree, Complete Binary Tree, Extended Binary Tree, Tree Traversal algorithms: In-order, Pre-order and Post-order. Constructing Binary Tree from given Tree Traversal, Binary Heaps, Heap Operations, Threaded Binary trees, Traversing Threaded Binary trees, AVL Tree, B-Tree.	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (31 to 40)	**CO4**
UNIT 5 **Graphs**	**Graphs**	Graphs: Terminology used with Graph, Graph Sorting Techniques: Representations: Adjacency matrices, Adjacency List. Connected Component, Spanning Trees, Prim's and Kruskal's algorithm, Shortest Path algorithms: Dijkstra Algorithm,Floyd Warshall's Algorithm	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (41 to 53)	**CO5**
	Hashing	Sorting Algorithms. Hashing: Hash Functions, Collision-Resolution Techniques.				
Total				**40L+20P**		

Textbooks

Sr No	**Book Details**
1.	Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data Structures and Algorithms in Python(An Indian Adaptation)", Wiley Publication
2.	Dr Shriram K. Vasudevan (Author), Mr Abhishek S. Nagarajan (Author), Prof Karthick Nanmaran (Author) "DATA STRUCTURES USING PYTHON" 12 March 2021, Oxford Higher Education, First Edition

3.	Hemant Jain “Problem Solving in Data Structures & Algorithms Using Python” 1 January 2022, Third Edition
Reference Books	
Sr No	**Book Details**
1.	Kiran Gurbani, Krupa Kamdar “Data Structure (Mumbai University), Himalaya Publishing House.
2.	Harsh Bhasin (Author) “Data Structures with Python: Get familiar with the common
3.	Data Structures and Algorithms in Python”, 1 May 2023, BPB Publication.
4.	Sanjay Patidar Upendra Singh Sumit Kumar Sharma “DATA STRUCTURES AND ALGORITHMS USING PYTHON “ 13 April 2023, Notion Press
Links: NPTEL/ YouTube/ Faculty Video Link	
https://nptel.ac.in/courses/106/106/106106127/ https://www.youtube.com/watch?v=zWg7U0OEAoE&list=PLBF3763AF2E1C572F	
https://www.youtube.com/watch?v=4OxBvBXon5w&list=PLBF3763AF2E1C572F&index=22	
https://www.youtube.com/watch?v=cR4rxllyiCs&list=PLBF3763AF2E1C572F&index=23 https://nptel.ac.in/courses/106/106/106106127/	
https://www.youtube.com/watch?v=9zpSs845wf8&list=PLBF3763AF2E1C572F&index=24	
https://www.youtube.com/watch?v=hk5rQs7TQ7E&list=PLBF3763AF2E1C572F&index=25	
https://www.youtube.com/watch?v=KW0UvOW0XIo&list=PLBF3763AF2E1C572F&index=5	

Subject Name: Accounting and Financial Management	**L-T-P [3-1-0]**
Subject Code: BMICA0301	**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: Basic knowledge of finance and accounting.	
Course Objective: This course aims to develop cognizance about the importance of accounting in organization financial statements. Students will become acquainted with the mechanics of financial statement preparation, understanding corporate financial statements, their analysis and interpretation, the significance of IFRS in accounting discipline, and the notion of management quality analysis and wealth generation.	

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Understand various Basics of Accounting and Balancing of Accounts	**K1**
CO2	Understand Accounting Standards for Routine Business Transactions	**K2**
CO3	Prepare and Present financial statements.	**K3**
CO4	Understand and apply accounting techniques for Analysis	**K3, K4**
CO5	Understand flow of Cash and Working Capital Management	**K1, K3**

Syllabus

Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment / Lab Nos**	**CO Mapping**
UNIT 1 **Introduction to Accounting**	**Overview**	Accounting Concepts, Conventions, and Principles; Meaning and Scope of Accounting: Evolution and Users of Accounting, Types of Accounting, Basic Accounting	Lectures, PPTs, Notes	8L	Assignment	**CO1**

		terminologies, Accounting Equation; Artificial Intelligence in Accounting.				
UNIT 2 **Mechanics of Accounting**	**Accounting principles and standards**	Matching of Indian Accounting Standards with International Accounting Standards, IAS, IFRS, Ind AS Double entry system of Accounting, Golden Rules of Accounting, journalizing of transactions; Ledger posting and Trial Balance.	Lectures, PPTs, Notes	8L	Assignment	**CO2**
UNIT 3 **Preparation of Financial Statements**	**Financial Statements**	Financial Statements and its presentation, Preparation of Income Statement and Balances sheet as per companies Act 2013,Use of Excel in preparation of Balance sheet.	Lectures, PPTs, Notes	8L	Assignment	**CO3**
UNIT 4 **Financial Statement Analysis**	**Analysis of financial statement:**	Ratio Analysis- solvency ratios, profitability ratios, activity ratios, liquidity ratios, market capitalization ratios ; Common Size Statement ; Comparative Balance Sheet and Trend Analysis.	Lectures, PPTs, Notes	8L	Assignment	**CO4**
UNIT 5 **Working Capital Management**	**Working Capital Management**	Concept of Gross and Net Working Capital, Preparation of Schedule of Changes in Working Capital, Cash Flow Statement: Various cash and non-cash transactions, flow of cash, preparation of Cash Flow Statement and its analysis.	Lectures, PPTs, Notes	8L	Assignment	**CO5**
Total				**40L**		

Textbooks	
Sr No	**Book Details**
1.	Maheshwari, S. N., & Maheshwari, S. K. (2001). Advanced Accountancy Volume-I. Vikas Publishing House.
2.	Tulsian, P. C. (2002). Financial Accounting. Pearson Education India.
Reference Books	
Sr No	**Book Details**

1.	Gupta, K. (2011). Khan, MY and Jain, PK, Financial Management: Text, Problems and Case. Journal ofServices Research, 11(2).
2.	Hasan, A. R. (2021). Artificial Intelligence (AI) in accounting & auditing: A Literature review. Open Journal of Businessand Management, 10(1), 440-465.
3.	Schroeder, R. G., Clark, M. W., & Cathey, J. M. (2022). Financial accounting theory and analysis: text and cases. JohnWiley & Sons.
4.	Collier, P. M. (2015). Accounting for managers: Interpreting accounting information for decision making. John Wiley &Sons
5.	Gupta, K. (2011). Khan, MY and Jain, PK, Financial Management: Text, Problems and Case. Journal of ServicesResearch, 11(2).
6.	Hasan, A. R. (2021). Artificial Intelligence (AI) in accounting & auditing: A Literature review. Open Journal of Business and Management, 10(1), 440-465.

Links: NPTEL/ YouTube/ Faculty Video Link

https://abmagazine.accaglobal.com/middle-east-south-asia/en.html

https://www.icai.org/category/e-journal

https://www.accountingseed.com/resource/blog/learn-accounting-for-free-with-these-resources/

https://www.journalofaccountancy.com/news.html

Subject Name: Emerging Trends & Technologies	**L-T-P [4-0-0]**
Subject Code: BMICA0303	**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: The student must understand basic computer terminology. The student must have knowledge of some programming language.	
Course Objective: Provide an understanding of the role computation can play in solving problems. Position students so that they can compete for research projects and excel in subjects with programming components.	

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Demonstrate the knowledge of the basic structure, components, Features and generations of computers.	**K3**
CO2	Compare and contrast features, functioning & types of operating system and computer networks.	**K4**
CO3	Demonstrate architecture, functioning & services of the Internet and basics of multimedia.	**K3**
CO4	Implement the working concepts of MS-Office	**K3**
CO5	Illustrate the emerging trends and technologies in the field of Information Technology.	**K3**

Syllabus

Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**
UNIT 1 Introduction to Computer	**Introduction to Computer**	Definition Computer: Computer Hardware & Computer Software Components: Hardware – Introduction, Input devices, Output devices, Central Processing Unit Memory – Primary and Secondary Software – Introduction, Types– System and Application Computer Languages Introduction, Concept of Compiler, Interpreter & Assembler Algorithms – Introduction, Definition,	Lectures, PPTs, Notes	8L	Assignment	**CO1**

		Characteristics, Limitations, Conditions in pseudo-code, Loops in pseudo code.				
UNIT 2 Operating system	**Operating system**	Operating system Definition, Functions, Types, Classification, Elements of command based and GUI based operating system. Windows Operating System Commands Computer Network: Overview, Standalone, Types (LAN, WAN and MAN), Data communication, topologies.	Lectures, PPTs, Notes	8L	Assignment	**CO2**
UNIT 3 Internet	**Internet**	Internet: Overview, Architecture, Functioning, Basic services like WWW, FTP, Telnet, Gopher etc., Search engines, E-mail, Web Browsers. Internet of Things (IoT) Definition, Sensors, their types and features, Smart Cities, Industrial Internet of Things.	Lectures, PPTs, Notes	8L	Assignment	**CO3**
UNIT 4 MS-Office	**MS-Office**	MS-Office: Basic Concepts, Features, Applications and handling of MS-Word, MS-PowerPoint and MS-Excel.	Lectures, PPTs, Notes	8L	Assignment	**CO4**
UNIT 5 Emerging Technologies	**Emerging Technologies**	Emerging Technologies: Introduction, overview, features, limitations and application areas of Cloud Computing, Big data, Grid Computing, Artificial Intelligence and Virtual Reality.	Lectures, PPTs, Notes	8L	Assignment	**CO5**
Total				**40L**		

Textbooks	
Sr No	**Book Details**
1.	Raja Raman V.(2003), “Fundamentals of Computers”, Prentice-Hall of India.
2.	Norton P.(2017), “Introduction to Computers”, McGraw Hill Education.
3.	Goel A.(2010), “Computer Fundamentals”, Pearson.

Reference Books	
Sr No	**Book Details**
1.	Balagurusamy E.(2020),“Fundamentals of Computers”, McGraw-Hill.
2.	Thareja R.(2023),“Fundamentals of Computers”, Oxford University Press.
3.	Bindra J.,“The Tech Whisperer-on Digital Transformation and the Technologies that Enableit” Penguin

Links: NPTEL/ YouTube/ Faculty Video Link
https://www.youtube.com/watch?v=eEo_aacpwCw
https://www.youtube.com/watch?v=WJ-UaAaumNA
https://www.youtube.com/watch?v=cNwEVYkx2Kk
https://www.youtube.com/watch?v=W3yttwGE-C0
https://www.youtube.com/watch?v=yCVy5Kw0l8s

Subject Name: Advanced Python						**L-T-P [0-0-8]**
Subject Code: BMICA0355					**Applicable in Department: MCA Integrated**	
Pre-requisite of Subject: Basic knowledge of Python programming.						
Course Objective: To become familiar with Python's Object-Oriented Concepts, functional programmingand create GUI application and to gain the knowledge of Python libraries.						
Course Outcomes (CO)						
Course outcome: After completion of this course students will be able to:						**Bloom's Knowledge Level(KL)**
CO1	Describe OOPs concepts in Python.					K2
CO2	Explain functional and GUI programming.					K5
CO3	Discuss Python libraries for data handling.					K2
CO4	Analyze data using visualization libraries.					K4
CO5	Examine web scraping application for real world data.					K4
Syllabus						
Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**

UNIT-1 Classes and Objects	**Classes and Objects**	Introduction of Python Classes and objects, User-Defined Classes, Class Variables and Instance Variables, Instance methods, Class method, static methods, constructor in python, parametrized constructor, Magic Methods in python, Object as an argument, Instances as Return Values, namespaces, Introduction to inheritance and polymorphism, Abstract Class, Introduction to Abstraction and Encapsulation.	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment	4L+10P	**Assignment, Lab (1 to 35)**	**CO1**
UNIT-2 Functional and GUI Programming	**Functional and GUI Programming**	Introduction to GUI Programming, Settling widgets in the window's interior, Numeric Widgets, Boolean Widgets, Selection Widgets, String Widgets, Date Picker, Color Picker, Container Widgets, Comparison of Array, List and Linked list, Types of linked list: Singly Linked List, DoublyLinked List, Circular Linked List, Polynomial Representation and Addition of Polynomials. Creating a GUI Application, Tkinter, button, canvas.	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment	4L+10P	**Assignment, Lab (36 to 80)**	**CO2**
	GUI Programming	Basic Operation, Indexing, slicing and Iterating, Multidimensional arrays, NumPy Data types, Reading and writing data on Files.				
UNIT-3 Libraries for Data Handling	**NumPy**	Basic Operation, Indexing, slicing and Iterating, Multidimensional arrays, NumPy Data types, Reading and writing data on Files.	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment	4L+10P	**Assignment, Lab (81 to 120)**	**CO3**
	Pandas	Series and Data Frames, Grouping, aggregation, Merge Data Frames, Generate summary tables, Group data into logical pieces, Manipulation of data.				
UNIT-4 Libraries in	**Matplotlib**	Scatter plot, Bar charts, histogram, Stack charts, Legend title Style, Figures and subplots, plotting function in pandas, Labelling and arranging figures, Save plots.	Lectures, Code Walkthroughs, Hand-on	4L+10P	**Assignment, Lab**	**CO4**

Data Visualization	**Seaborn**	style function, color palettes, heatmaps, distribution plots, category plot, regression plot.	Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment.		**(121 to 169)**	
	Plotly	Lineplots, Areaplots, Scatterplots, Bubbleplots , Stacked bar charts, Grouped bar charts, Pie charts, Tables, Dashboards.				
UNIT-5 Web Scraping with Python	**Web Scraping**	Introduction, Web Crawling v/s Web Scraping, Uses of Web Scraping, Components of a WebScraper, working of a Web Scraper, Crawl, Parse and Transform Store the Data.	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment.	4L+10P	**Assignment, Lab (170 to 305)**	**CO5**
	Beautiful Soup	Introduction to Beautiful Soup library, Accessing Tags, Navigable Strings, Navigating and searching with Beautiful Soup,Web Scraping.				
	Example	Scraping Flipkart Website, Introduction to Github.				
Total				**20L+50P**		

Subject Name: Advanced Python	**L-T-P [0-0-8]**

Course Objective: Develop proficiency in Python programming for effective problem-solving. Cover algorithms, data structures, debugging, and application across domains, emphasizing efficiency, documentation, and ethical considerations.

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO 1	Implement OOPs concepts in Python.	**K3**
CO2	Create functional and GUI programming.	**K5**
CO3	Demonstrate the use of Python libraries for data handling.	**K3**
CO4	Construct visualization libraries concepts in python.	**K5**
CO5	Apply web scraping application for data extraction.	**K3**

List of Practical

Sr No	**Program Title**	**CO Mapping**
1	Write a program illustrating class definition and accessing class members.	**CO1**
2	Write a program to implement default constructor, parameterized constructor, and destructor.	**CO1**
3	Create a Python class named Rectangle constructed by a length and width. a. Create a method called area which will compute the area of a rectangle.	**CO1**
4	Create a class called Numbers, which has a single class attribute calledMULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers). a. Write an instance method called add which returns the sum of theattributes x and y. Write a class method called multiply, which takes a single numberparameter a and returns the product of a and MULTIPLIER.	**CO1**

5	Create a class named as Student to store the name and marks in threesubjects. Use List to store the marks. a. Write an instance method called compute to compute total marks andaverage marks of a student. Write a method called display to display student information.	**CO1**
6	Create a Python class named Circle constructed by a radius. Use a classvariable to define the value of constant PI. a. Write two methods to be named as area and circum to compute the area and the perimeter of a circle respectively by using class variable PI. b. Write a method called display to print area and perimeter.	**CO1**
7	Write a program that has a class called Fraction with attributes numeratorand denominator. a. Write a method called get data to enter the values of the attributes. Write a method show to print the fraction in simplified form.	**CO1**
8	Write a program that has a class Numbers with a list as an instancevariable. a. Write a method called insert_element that takes values from user. Write a class method called find_max to find and print largest value inthe list.	**CO1**
9	Create a class called Complex. Write a menu driven program to read, display, add and subtract two complex numbers by creating correspondinginstance methods.	**CO1**
10	Write a program that has a class Point with attributes x and y. a. Write a method called midpoint that returns a midpoint of a line joiningtwo points. b. Write a method called length that returns the length of a line joining two points.	**CO1**
11	Create a class called Complex. Write a menu driven program to read, display, add and subtract two complex numbers by creating corresponding instance methods.	**CO1**
12	Write a Python program to create a class called "Rectangle" with attributes length and width. Include methods to calculate the perimeterand area of the rectangle.	**CO1**
13	Implement a Python class called "Bank Account" with attributes account number, account holder name, and balance. Include methods to depositand withdraw money from the account.	**CO1**
14	Write a Python program to create a class called "Student" with attributes roll number, name, and marks in three subjects. Include a method tocalculate the average marks of the student.	**CO1**
15	Implement a Python class called "Car" with attributes make, model, and year. Include methods to start the car, stop the car, and display its details.	**CO1**
16	Write a Python program to create a class called "Book" with attributes title, author, and price. Include methods to calculate the discounted price of the book based on a discount percentage provided.	**CO1**
17	Implement a Python class called "Bank" with attributes bank name and branch. Include methods to add a new account, display all accounts, andsearch for an account based on the account number.	**CO1**

18	Write a Python program to create a class called "Rectangle" with attributes length and width. Include a method to check if the rectangle is a square or not.	**CO1**
19	Implement a Python class called "Employee" with attributes name, designation, and experience. Include methods to promote an employee toa higher designation based on their experience.	**CO1**
20	Write a Python program to create a class called "Employee" with attributes name, employee ID, and salary. Include a method to display the employee details.	**CO1**
21	Write a program to illustrate the use of following built-in methods: a. hasattr(obj,attr) b. getattr(object, attribute_name [, default]) c.setattr(object, name, value) d. delattr(class_name, name)	**CO1**
22	Write a Program to illustrate the use of __str__(), __repr__(), __new__, __doc__, __dict__, __name__and__bases__methods.	**CO1**
23	Write a program to create class Employee. Display the personal information and salary details of 5 employees using single inheritance.	**CO1**
24	WAP that extends the class Employee. Derive two classes Manager andTeam Leader from Employee class. Display all the details of the employee working under a particular Manager and Team Leader.	**CO1**
25	Write a program that has a class Point. Define another class Location which has two objects (Location and destination) of class Point. Also,define a function in Location that prints the reflection on the y-axis.	**CO1**
26	Write a program that create a class Distance with member's km and metres. Derive classes School and office which store the distance fromyour house to school and office along with other details.	**CO1**
27	Write a program to create an abstract class Vehicle. Derive three classes Car, Motorcycle and Truck from it. Define appropriate methods and print the details of vehicle.	**CO1**
28	Write a program to demonstrate hybrid inheritance and show MRO for each class.	**CO1**
29	Write a program to overload + operator to multiply to fraction object of fraction class which contain two instance variable numerator and denominator. Also, define the instance method simplify() to simplify the fraction objects.	**CO1**
30	Write a program to compare two-person object based on their age by overloading > operator.	**CO1**
31	Write a program to overload in operator.	**CO1**
32	WAP to create a Complex class having real and imaginary as it attributes. Overload the +,-,/,* and += operators for objects of Complex class	**CO1**
33	WAP to Show the concept of inner function.	**CO2**
34	WAP to create a decorator which will convert a string into upper case string.	**CO2**
35	WAP to show the concept of nested decorator.	**CO2**
36	WAP to decorate a function with arguments.	**CO2**

37	WAP to decorate instance method	**CO2**
38	WAP to calculate sum of 1,2,3,4,5 using reduce function.	**CO2**
39	WAP to generate numbers from 1 to 10 using generator.	**CO2**
40	WAP to decide number is even or odd using generator.	**CO2**
41	WAP to generate square of 1,2,3,4,5,6,7,8,9,10 using generator.	**CO2**
42	WAP to generate square of even number up to 10 using generator andsave in list.	**CO2**
43	WAP to make a co-routine which will print all name with prefix Dear.	**CO2**
44	WAP to close a co-routine.	**CO2**
45	WAP to iterate tuple using iter() and next() method.	**CO2**
46	WAP to iterate a string using iter and next method.	**CO2**
47	WAP to print numbers from 1 to 20 using iterator and generate Stop Iteration exception once we reach limit.	**CO2**
48	Hello World: Display a simple "Hello, World!" message box.	**CO2**
49	Button: Create a button that displays a message when clicked.	**CO2**
50	Entry: Create a text entry field and display the entered text.	**CO2**
51	Check button: Create a checkbox and display the selected options	**CO2**
52	Radio button: Create radio buttons and display the selected option.	**CO2**
53	List box: Create a list box and display the selected items.	**CO2**
54	Text: Create a text area and display the entered text.	**CO2**
55	Menu: Create a menu with different options.	**CO2**
56	Message: Display a message in a dialog box.	**CO2**
57	Progress bar: Create a progress bar that updates over time python	**CO2**
58	Scale: Create a scale widget and display the selected value.	**CO2**

59	Spin box: Create a spin box and display the selected value.	**CO2**
60	Canvas: Create a canvas and draw shapes on it.	**CO2**
61	Label Frame: Create a labeled frame with widgets inside.	**CO2**
62	Scrollbar: Add a scrollbar to a widget like a text area or list box	**CO2**
63	Frame: Create a frame and place widgets inside it.	**CO2**
64	Tree view: Create a tree view widget to display hierarchical data	**CO2**
65	Notebook: Create a notebook widget with tabs.	**CO2**
66	File Dialog: Open a file dialog to select a file.	**CO2**
67	Color Dialog: Open a color dialog to select a color.	**CO2**
68	Button Counter: Create a button that increments a counter when clicked.	**CO2**
69	Checkbox List: Display a list of checkboxes and show selected options.	**CO2**
70	Dropdown Menu: Create a dropdown menu with multiple options.	**CO2**
71	Slider Value Display: Display the current value of a slider widget.	**CO2**
72	Text Input and Button: Take user input in a text box and display it when a button is clicked.	**CO2**
73	Radio Buttons: Present a set of options as radio buttons and display the selected option.	**CO2**
74	Progress Bar: Show the progress of a task using a progress bar widget.	**CO2**
75	Password Input: Create a password input field that hides the entered characters.	**CO2**
76	File Uploader: Enable users to upload files and display the selected file name.	**CO2**
77	Creating Arrays: Create NumPy arrays using various methods like np.array(), np.zeros(), np.ones(), np.arange(), etc.	**CO3**
78	Array Shape and Size: Get the shape and size of a NumPy array using the shape and size attributes. NumPy	**CO3**
79	Array Indexing: Access and modify individual elements of a NumPy array using indexing	**CO3**
80	Array Slicing: Extract a subset of elements from a NumPy array using slicing.	**CO3**

81	Array Reshaping: Change the shape of a NumPy array using the reshape() function.	**CO3**
82	Array Arithmetic: Perform basic arithmetic operations (addition, subtraction, multiplication, division) on NumPy arrays.	**CO3**
83	Array Broadcasting: Perform element-wise operations on arrays with different shapes using broadcasting rules.	**CO3**
84	Array Aggregation: Calculate aggregate values on arrays, such as sum(), min(), max(), mean(), etc. using NumPy	**CO3**
85	Array Transposition: Transpose a NumPy array using the transpose() function.	**CO3**
86	Write a program that demonstrates advanced array indexing techniques,such as indexing with boolean arrays or using fancy indexing to select specific elements or subsets of an array.	**CO3**
87	Write a program using NumPy to perform data manipulation tasks, such as sorting arrays, removing duplicates, or finding unique elements in anarray.	**CO3**
88	Array Sorting: Sort the elements of a NumPy array using the sort() function.	**CO3**
89	Array Filtering: Filter elements in a NumPy array based on a condition using Boolean indexing.	**CO3**
90	Array Statistics: Calculate statistical measures like mean, median, standard deviation using functions like np.mean(), np.median(), np.std().	**CO3**
91	Array Randomization: Generate random numbers or arrays using functions from the np.random module.	**CO3**
92	Array Dot Product: Compute the dot product of two NumPy arrays using the dot() function.	**CO3**
93	Array Matrix Operations: Perform matrix operations like matrix multiplication, matrix inverse using functions from the np.linalg module.	**CO3**
94	Array File I/O: Save and load NumPy arrays from files using functions like np.save() and np.load().	**CO3**
95	Array Masking: Create a mask array to select or manipulate specific elements of a NumPy array based on a condition.	**CO3**
96	Array Broadcasting: Understand and utilize broadcasting rules in NumPy for efficient computations.	**CO3**
97	Write a program to finds the cube root of values using scipy library.	**CO3**
98	Read and Load a CSV File into a Pandas DataFrame using pandas.read_csv.	**CO3**
99	Access and Display the First N Rows of a DataFrame using DataFrame.head(N).	**CO3**
100	Access and Display the Last N Rows of a DataFrame using DataFrame.tail(N).	**CO3**
101	Retrieve Basic Information about a DataFrame using DataFrame.info.	**CO3**

102	Perform Descriptive Statistics on a DataFrame using DataFrame.describe.	**CO3**
103	Filter Rows of a DataFrame based on a Condition using Boolean Indexing.	**CO3**
104	Rename Columns in a DataFrame using DataFrame.rename.	**CO3**
105	Group Data in a DataFrame using DataFrame.groupby.	**CO3**
106	Perform Aggregation on Grouped Data using GroupBy.agg.	**CO3**
107	Sort a DataFrame by One or Multiple Columns using DataFrame.sort_values.	**CO3**
108	Perform Basic Arithmetic Operations on Columns of a DataFrame.	**CO3**
109	Apply a Function to Each Element or Column of a DataFrame using DataFrame.apply or Data Frame.applymap.	**CO3**
110	Reshape Data using Pivot Tables using Data Frame.pivot_table.	**CO3**
111	Perform Data Visualization using pandas.plotting or matplotlib.pyplot.	**CO3**
112	Create a Simple Line Plot using matplotlib.pyplot.plot.	**CO4**
113	Create a Scatter Plot using matplotlib.pyplot.scatter.	**CO4**
114	Create a Bar Chart using matplotlib.pyplot.bar.	**CO4**
115	Create a Histogram using matplotlib.pyplot.hist.	**CO4**
116	Create a Pie Chart using matplotlib.pyplot.pie.	**CO4**
117	Create a Box Plot using matplotlib.pyplot.boxplot.	**CO4**
118	Create a Heatmap using matplotlib.pyplot.imshow.	**CO4**
119	Customize Plot Labels and Titles using matplotlib.pyplot.xlabel, matplotlib.pyplot.ylabel, and matplotlib.pyplot.title.	**CO4**
120	Customize Plot Colors, Line Styles, and Marker Styles using matplotlib.pyplot.plot parameters.	**CO4**
121	Add Gridlines to a Plot using matplotlib.pyplot.grid.	**CO4**
122	Add Legends to a Plot using matplotlib.pyplot.legend.	**CO4**
123	Create Subplots using matplotlib.pyplot.subplots.	**CO4**

124	Save a Plot as an Image File using matplotlib.pyplot.savefig.	**CO4**
125	Create 3D Plots using mpl_toolkits.mplot3d module.	**CO4**
126	Create Error Bars on a Plot using matplotlib.pyplot.errorbar.	**CO4**
127	Customize Axis Ticks and Tick Labels using matplotlib.pyplot.xticks and matplotlib.pyplot.yticks.	**CO4**
128	Create a Bar Plot with Stacked Bars using matplotlib.pyplot.bar and the bottom parameter.	**CO4**
129	Create a Scatter Plot using seaborn.scatterplot.	**CO4**
130	Create a Line Plot using seaborn.lineplot.	**CO4**
131	Create a Bar Plot using seaborn.barplot.	**CO4**
132	Create a Histogram using seaborn.histplot.	**CO4**
133	Create a Box Plot using seaborn.boxplot.yh9.**	**CO4**
134	Create a Violin Plot using seaborn.violinplot.	**CO4**
135	Create a Heatmap using seaborn.heatmap.	**CO4**
136	Create a Pair Plot using seaborn.pairplot.	**CO4**
137	Create a Joint Distribution Plot using seaborn.jointplot.	**CO4**
138	Create a KDE (Kernel Density Estimate) Plot using seaborn.kdeplot.	**CO4**
139	Create a Categorical Scatter Plot using seaborn.stripplot.	**CO4**
140	Create a Categorical Bar Plot using seaborn.countplot.	**CO4**
141	Create a Facet Grid using seaborn.FacetGrid.	**CO4**
142	Customize Plot Colors and Styles using seaborn.set_palette and seaborn.set_style.	**CO4**
143	Add Error Bars to a Plot using seaborn.barplot or seaborn.pointplot with the ci parameter.	**CO4**
144	Create a Clustered Heatmap using seaborn.clustermap.	**CO4**
145	Create a Regression Plot using seaborn.regplot.	**CO4**

146	Create a Pairwise Relationship Plot using seaborn.pairplot or seaborn.scatterplot with multiple variables.	**CO4**
147	Create a Boxen Plot using seaborn.boxenplot.	**CO4**
148	Create a Stacked Bar Plot using seaborn.barplot with the hue parameter.	**CO4**
149	Write a program to draw a line chart using Plotly	**CO4**
150	Write a program to draw a Bar chart using Plotly	**CO4**
151	Write a program to draw a Histogram chart using Plotly	**CO4**
152	Write a program to draw a scatter plot using Plotly	**CO4**
153	Write a program to draw a Bubble chart using Plotly	**CO4**
154	Write a program to draw a pie chart using Plotly	**CO4**
155	Write a program to draw a Boxplot using Plotly	**CO4**
156	Write a program to draw a Violin Plots using Plotly	**CO4**
157	Write a program to draw a Gant chart using Plotly	**CO4**
158	Write a Python program to find the title tags from a given html document.	**CO5**
159	Write a Python program to retrieve all the paragraph tags from a given html document.	**CO5**
160	Write a Python program to get the number of paragraph tags of a given html document.	**CO5**
161	Write a Python program to extract the text in the first paragraph tag of a given html document.	**CO5**
162	Write a Python program to find the length of the text of the first <h2> tag of a given html document.	**CO5**
163	Write a Python program to find the text of the first <a> tag of a given html text.	**CO5**
164	Write a Python program to find the href of the first <a> tag of a given html document.	**CO5**
165	Write a Python program to a list of all the h1, h2, h3 tags from the webpage python.org.	**CO5**
166	Write a Python program to extract all the text from a given web page.	**CO5**
167	Write a Python program to print the names of all HTML tags of a given web page going through the document tree.	**CO5**

168	Write a Python program to retrieve children of the html tag from a given web page.	**CO5**
169	Write a Python program to retrieve all descendants of the body tag from given web page.	**CO5**
170	Write a Python program to print content of elements that contain a specified string of a given web page.	**CO5**
171	Write a Python program to print the element(s) that has a specified id of a given web page.	**CO5**
172	Write a Python program to create a Beautiful Soup parse tree into a nicely matted Unicode string, with a separate line for each HTML/XML tagand string.	**CO5**
173	Write a Python program to find the first tag with a given attribute value in an html document.	**CO5**
174	Write a Python program to find tag(s) beneath other tag(s) in a given html document.	**CO5**
175	Write a Python program to find tag(s) directly beneath other tag(s) in a given html document.	**CO5**
176	Write a Python program to find the siblings of tags in a given html document.	**CO5**
177	Write a Python program to find tags by CSS class in a given html document.	**CO5**
178	Write a Python program to change the tag's contents and replace with the en string.	**CO5**
179	Write a Python program to add to a tag's contents in a given html documents.	**CO5**
180	Write a Python program to insert a new text within a url in a specified position	**CO5**
181	Write a Python program to insert tags or strings immediately before specified tags or strings.	**CO5**
182	Write a Python program to insert tags or strings immediately after specified tags or strings.	**CO5**
183	Write a Python program to remove the contents of a tag in a given html document.	**CO5**
184	Write a Python program to extract a tag or string from a given tree of html document.	**CO5**
185	Write a Python program to remove a tag from a given tree of html document and destroy it and its contents.	**CO5**
186	Write a Python program to remove a tag or string from a given tree of html document and replace it with the given tag or string.	**CO5**
187	Write a Python program to wrap an element in the specified tag and create the new wrapper.	**CO5**
188	Write a Python program to replace a given tag with whatever's inside a given tag.	**CO5**

Required Software and Tools	
1. Jupiter Notebook	
Textbooks	
Sr No	**Book Details**
1.	Kriti Kumari Sinha , “Advanced Python Guide: Master concepts, build applications, and prepare for interviews Paperback”, 20 June 2024
2.	Martin C. Brown , “Python: The Complete Reference Paperback”, 20 March 2018
3.	Kenneth A. Lambert, ―Fundamentals of Python: First Programs, CENGAGE Learning, (2012)
Reference Books	
Sr No	**Book Details**
1.	John V Guttag, ―Introduction to Computation and Programming Using Python‘‘, Revised and expanded Edition, MIT Press , (2013)
2.	Charles Dierbach, ―Introduction to Computer Science using Python: A Computational Problem Solving Focus, Wiley India Edition, (2013).
3.	Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”, 2nd edition, Updated for Python 3, Shroff/O‘Reilly Publishers,(2016).
4.	Robert Sedgewick, Kevin Wayne, Robert Dondero: Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson IndiaEducation Services Pvt. Ltd.,(2016).
5.	Guido van Rossum and Fred L. Drake Jr, ―An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., (2011)
Links: NPTEL/ YouTube/ Faculty Video Link	
https://www.youtube.com/watch?v=Rq_3gA2h1RA	
https://www.youtube.com/watch?v=-GhzpvvIXlM&list=PLS1QulWo1RIY6fmY_iTjEhCMsdtAjgbZM **https://www.youtube.com/watch?v=rDj8EBv9ErA**	

https://www.youtube.com/watch?v=aYmcRnmZVGQ&list=PL9n0l8rSshSnragNblKDBsT8Xu3otp3jA
https://www.youtube.com/watch?v=9GYmFXBitBw&list=PLBSCvBlTOLa8rf2kGkP_Bx5xXqT-er4Yq
https://www.youtube.com/watch?v=XVv6mJpFOb0

Subject Name: Operating System Lab		**L-T-P [0-0-4]**
Subject Code: BMICA0354		**Applicable in Department: MCA Integrated**
Lab Experiments		
Course Objective: This course gives an ability to students to construct codes for OS API and basics of OS mechanisms and Hands-on and practical experience with usage of the Linux OS and basics of Shell Programming.		
Course Outcomes (CO)		
Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Implement Linux commands to understand the concept of virtualization	**K3**
CO2	Solve the real-world problems using shell programming and shell scripting.	**K3**
CO3	Analyze process management and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.	**K5**
CO4	Implement Process Synchronization and analyze deadlock handling techniques.	**K3**
CO5	Simulate the continuous and non-continuous memory allocation concepts and analyze disk scheduling algorithms.	**K3, K4**

List of Practicals

Sr No	Program Title	CO Mapping
1	**File Manipulation Commands:** cat command, cp command, ls command, rm command, mv command, mkdir command, rmdir command, find command, grep command, wc command, sort command, more command, head command and tail command.	**CO1**
2	**Status Alter Commands:** chgrp command, chown command & chmod command.	**CO1**
3	**Compile Commands:** cc command	**CO1**
4	**Process Commands:** ps command, kill command.	**CO1**
5	**Miscellaneous Commands:** Echo command, cal command, date command, whoami command, expr command & test	**CO1**

	command.	
6	**Filter Commands:** cut command, paste command, head command & tr command.	**CO1**
7	Write a program to create a child process and print the process ids of parent and child process.	**CO2**
8	Write a Shell program to check the given number is even or odd.	**CO2**
9	Write a Shell program to find the factorial of a number.	**CO2**
10	Write a Shell program to swap the two integers.	**CO2**
11	Write a shell script to calculate the gross salary if basic salary is given. DA is 40% of basic salary and HRA is 60% of basic salary.	**CO2**
12	Write a shell script to reverse the digits of a number.	**CO2**
13	Write a shell script to compute the sum of digits.	**CO2**
14	Write a shell script to convert the contents of files to uppercase, given multiple files on command line.	**CO2**
15	Write a shell script to print the just the time, just the day of month and just the day of week as desired by user.	**CO2**
16	Write a shell script to print a word n times. Taking the n and the word from command line.	**CO2**
17	Write a shell script to print the area and perimeter of a rectangle and area and circumference of a circle. The length, breadth and radius are inputted through keyboard.	**CO2**
18	Write a shell script to delete all lines containing the word “unix” from all files provided as command line arguments.	**CO2**
19	Write a shell script to search a number in the given list of numbers. Number is provided as first argument in command line arguments, the list of numbers follows after that.	**CO2**
20	Write a shell script to sort the numbers provided as command line arguments in a descending order.	**CO2**
21	Write a shell script to count the number of negative and positive numbers provided as command line arguments.	**CO2**
22	Write a shell Script to Concatenate two Strings.	**CO2**
23	Write a shell script to print Fibonacci series.	**CO2**
24	Write a shell script to find whether a year is leap year or not.	**CO2**
25	Write a shell script to find whether a number is prime or not.	**CO2**
26	Write a program Using First-Come, First-Served **(FCFS)** Scheduling to find average turnaround time and average	**CO3**

	waiting7time.	
27	Write a program Using **SJF** CPU scheduling algorithm to find average turnaround time and average waiting time.	**CO3**
28	Write a program Using **Priority** CPU scheduling algorithm to find average turnaround time and average waiting time.	**CO3**
29	Write a program Using **Round Robin scheduling** to calculate average turnaround time and average waiting time.	**CO3**
30	Write a Program to execute **Race Condition** of Process Synchronization.	**CO4**
31	Write a program to implement **Producer Consumer** Problem.	**CO4**
32	Write a Program to Implement **Dinning Philosophers** Problem.	**CO4**
33	Write a Program to Implement **Banker's Algorithm**.	**CO4**
34	Write a Program to Implement **FIRST FIT** Memory Allocation.	**CO5**
35	Write a Program to Implement **BEST FIT** Memory Allocation.	**CO5**
36	Write a Program to Implement **WORST FIT** Memory Allocation.	**CO5**
37	Write a Program to Implement **FIFO** Page Replacement Algorithm.	**CO5**
38	Write a Program to Implement **LRU** Page Replacement Algorithm.	**CO5**
39	Write a Program to Implement **Optimal** Page Replacement Algorithm.	**CO5**
40	Write a Program to Implement **FCFS** Disk Scheduling Algorithm.	**CO5**
41	Write a program to Implement **SSTF** Disk Scheduling Algorithm.	**CO5**
42	Write a Program to Implement **SCAN** Disk Scheduling Algorithm.	**CO5**
43	Write a Program to Implement **CSCAN** Disk Scheduling Algorithm.	**CO5**
44	Write a Program to Implement **LOOK** Disk Scheduling Algorithm.	**CO5**
45	Write a Program to Implement **CLOOK** Disk Scheduling Algorithm.	**CO5**
Required Software and Tools		
1. Linux (Open source) 2. Jupyter Notebook		

3. Anaconda
4. Numpy

Subject Name: Data Structures Lab		**L-T-P [0-0-4]**
Subject Code: BMICA0352		**Applicable in Department:** MCA Integrated
Lab Experiments		
Course Objective: Learn the basic concepts of algorithm analysis, along with implementation of linear and non-linear data structures.		
Course Outcomes (CO)		
Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Analyse systematic approach to organizing, writing and debugging Array programs	**K4**
CO2	Implement Stack and Queue	**K3**
CO3	Develop operations of linked list.	**K5**
CO4	Construct non-linear data structure operations.	**K5**
CO5	Implement sorting and searching algorithms using relevant data structures	**K3**

List of Practical

Sr No	Program Title	CO Mapping
1.	Construct a Code to find the maximum element in an array.	**CO1**
2.	Construct a Code to calculate the sum of all elements in an array.	**CO1**
3.	Construct a Code to reverse the elements of an array.	**CO1**
4.	Construct a Code to check if an array is sorted in ascending order.	**CO1**
5.	Construct a Code to count the occurrence of a specific element in an array.	**CO1**
6.	Construct a Code creation and traversal of 2D Array in row major and column majororder.	**CO1**
7.	Construct a code to print the transpose of a given matrix using function	**CO1**

8.	Program to find if a given matrix is Sparse or Not and print Sparse Matrix	**CO1**
9.	Construct a code to Implement Linear Search	**CO1**
10.	Construct a code to implement Binary Search	**CO1**
11.	Implementation of stack using a list	**CO2**
12.	Construct a python code to Infix to postfix conversion using a stack	**CO2**
13.	Construct a code for Balanced parentheses checker using a stack	**CO2**
14.	Implement Reverse a string using a stack.	**CO2**
15.	Implement Binary Search using Recursion.	**CO2**
16.	Construct a python program to print Fibonacci Series using Recursion.	**CO2**
17.	Queue implementation using a list	**CO2**
18.	Construct a code for Simulating a printer queue using a queue.	**CO2**
19.	Construct a code for Implementing a circular queue.	**CO2**
20	Implement queue using stack	**CO2**
21.	Create a single linked list and perform basic operations (insertion, deletion, traversal).	**CO3**
22.	Create a double linked list and perform basic operations (insertion, deletion, traversal).	**CO3**
23.	Create a circular linked list and perform basic operations (insertion, deletion, traversal).	**CO3**
24.	Reverse a single linked list.	**CO3**
25.	Check if a linked list is palindrome.	**CO3**
26	Reverse a double linked list.	**CO3**
27.	Find the middle element of a single linked list.	**CO3**
28.	Find the middle element of a double linked list.	**CO3**
29.	Merge two sorted single linked lists.	**CO3**

30.	Detect and remove a loop in a circular linked list.	**CO3**
31.	Construct a code to Insert, Delete and search and update a data in Binary Search Tree(BST)	**CO4**
32.	Construct a code for Tree Traversal (Preorder, Inorder, Postorder)	**CO4**
33.	Construct a code Count the number of Leaves in a Binary Tree	**CO4**
34.	Construct a code to find the Height of a Binary Tree	**CO4**
35.	Construct a code to print all Paths from the Root to Leaf Nodes in a Binary Tree	**CO4**
36.	Construct a code to convert a Binary Tree to its Mirror Tree	**CO4**
37.	Construct a code to find the Node with Minimum Value in a Binary Search Tree.	**CO4**
38.	Construct a code for Binary Search Tree (BST) Implementation	**CO4**
39.	A program to check if a Binary Tree is a Binary Search Tree (BST)	**CO4**
40.	Construct a code to check if a Binary Tree is a Balanced Binary Tree	**CO4**
41.	Construct a code to represent graph using adjacency matrix and adjacency list.	**CO5**
42.	Implement BFS and DFS algorithm.	**CO5**
43.	Implement the minimum cost spanning tree.	**CO5**
44.	Implement bubble sort in a non-recursive way.	**CO5**
45.	Implement selection sort in a non-recursive way.	**CO5**
46.	Implement insertion sort in a non-recursive way.	**CO5**
47.	Implement Merge sort in a non-recursive way.	**CO5**
48.	Implement Merge sort in a recursive way.	**CO5**
49.	Array-based Student Performance Analysis System	**CO5**
50.	Design a project based on stack data structure to create a web history checker.	**CO5**
51.	Design a dynamic Music Playlist using Linked List	**CO5**

52	Design Decision Tree Classifier for Disease Diagnosis using tree data structure.	**CO5**
53.	Design Road Network Navigation: Implementing a navigation system to find the shortest path between locations using road networks.	**CO5**
Required Software and Tools		
1. Vs Code 2. Jupiter Notebook		

Subject Name: Design Thinking-II	**L-T-P [2-1-0]**
Subject Code: BMICA0404	**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: Student must complete Design Thinking-I course.	
Course Objective: The objective of this course is to upgrade Design Thinking skills by learning & applying advanced and contextual Design Thinking Tools. It aims to solve a Real-Life Problem by applying Design Thinking to create an impact for all the stakeholders.	

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level (KL)**
CO1	Learn sophisticated design tools to sharpen their problem-solving skills.	**K2**
CO2	Construct innovate ideas using design thinking tools and converge to feasible idea for breakthrough solution	**K6**
CO3	Implement storytelling for persuasive articulation	**K3**
CO4	Understanding the nature of leadership empowerment	**K2**
CO5	Understand the role of a human being in ensuring harmony in society and nature	**K2**

Syllabus

Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**

UNIT-1 Introduction	**Design Thinking & Innovation**	Design Thinking Mindset and Principles, recap of 5-Step Process of Design Thinking, Design Approaches, additional in-depth examples of each design approaches. Simon Sinek's – Start with Why, The Golden Circle, Asking the "Why" behind each example (an in-class activity of asking 5-WHYS) The Higher Purpose, in-class activity for LDO & sharing insights Visualization and its importance in design thinking, reflections on wheel of life (in-class activity for visualization & Wheel of Life), Linking it with Balancing Priorities (in class activity), DBS Singapore and Bank of Americas' Keep the Change Campaign. Litter of Light & Arvind Eye Care Examples, understanding practical application of design thinking tools and concepts, case study on McDonald's Milkshake / Amazon India's Rural Ecommerce & Gillette. Working on 1-hour Design problem, Applying RCA and Brainstorm on innovative solutions. Main project allocation and expectations from the project.	Lectures, PPTs, Notes	**8L**	**Assignment**	**CO1**
UNIT-2 Refinement and prototyping	**Refine and narrow down**	Refine and narrow down to the best idea, 10-100-1000gm, QBL, Design Tools for Convergence – SWOT Analysis for 1000gm discussion. In-class activity for 10-100-1000gm & QBL.	Lectures, PPTs, Notes	**8L**	**Assignment**	**CO2**
	Prototyping (Convergence)	Prototyping mindset, tools for prototyping – Sketching, paper models, pseudo-codes, physical mockups, Interaction flows, storyboards, acting/role-playing etc, importance of garnering user feedback for revisiting Brainstormed ideas, Napkin Pitch, Usability, Minimum Viable Prototype, Connecting Prototype with 3 Laws, A/B Testing, Learning Launch.				
	Decision Making Tools and Approaches	Vroom Yetton Matrix, Shift-Left, Up, Right, Value Proposition, Case study: Career buddy, You-Me-Health Story & IBM Learning Launch. In-class activities on prototyping- paper-pen / physical prototype/ digital prototype of project's 1000gm idea				

UNIT-3 Storytelling, testing and assessment	**Storytelling**	Elements of storytelling, Mapping personas with storytelling, Art of influencing, Elevator Pitch, Successful Campaigns of well-known examples, in-class activity on storytelling. Testing of design with people, conducting usability test, testing as hypothesis, testing as empathy, observation and shadowing methods, Guerrilla Interviews, validation workshops, user feedback, record results, enhance, retest, and refine design, Software validation tools, design parameters, alpha & beta testing, Taguchi, defect classification, random sampling. Final Project Presentation and assessing the impact of using design thinking	Lectures, PPTs, Notes	**8L**	**Assignment**	**CO3**
UNIT-4 Innovation, quality and leadership	**Innovation**	Innovation: Need & Importance, Principles of innovations, Asking the Right Questions for innovation, Rationale for innovation, Quality: Principles & Philosophies, Customer perception on quality, Kaizen, 6 Sigma. FinTech case study of Design Thinking application – CANVAS Leadership, types, qualities and traits of leaders and leadership styles, Leaders vs Manager, Personas of Leaders & Managers, Connecting Leaders-Managers with 13 Musical Notes, Trait theory, LSM (Leadership Situational Model), Team Building Models: Tuckman's and Belbin's. Importance of Spatial elements for innovation	Lectures, PPTs, Notes.	**8L**	**Assignment**	**CO4**
UNIT-5 Understanding human desirability	**Comprehensive human goal**	Comprehensive human goal: the five dimensions of human endeavour (Manaviya - Vyavstha) are: Education- Right living (Sikhsa- Sanskar), Health – Self-regulation (Swasthya - Sanyam), Justice – Preservation (Nyaya- Suraksha), Production – Work (Utpadan – Karya), Exchange – Storage (Vinimya – Kosh), Darshan-Gyan-Charitra (Shifting the Thinking) Interconnectedness and mutual fulfilment among the four orders of nature recyclability and self-regulation in nature	Lectures, PPTs, Notes	**8L**	**Assignment**	**CO5**

	Thinking expansion for harmony	Self-exploration (Johari's window), group behaviour, interpersonal behaviour and skills, Myers-Briggs personality types (MBTI), FIRO-B test to repair relationships.				
Total				**40L**		

Textbooks	
Sr No	**Book Details**
1.	Arun Jain, UnMukt: Science & Art of Design Thinking, 2020, Polaris "Operating System: A Design- oriented Approach", Charles Patrick Crowley, 9th Edition, 2017
2.	Gavin Ambrose and Paul Harris, Basics Design 08: Design Thinking, 2010, AVA Publishing SA "Design of the UNIX Operating Systems", Maurice J. Bach., 1st Edition, 2015
3.	R R Gaur, R Sangal, G P Bagaria, A Foundation Course in Human Values and Professional Ethics, First Edition, 2009, Excel Books: New Delhi.
Reference Books	
Sr No	**Book Details**
1.	Jeanne Liedta, Andrew King and Kevin Benett, Solving Problems with Design Thinking – Ten Stories of What Works, 2013, Columbia Business School Publishing
2.	Dr Ritu Soryan, Universal Human Values and Professional Ethics, 2022, Katson Books
3.	Vijay Kumar, 101 Design Methods: A Structured Approach for Driving Innovation in Your Organization, 2013, John Wiley and Sons Inc, New Jersey
4.	Roger L. Martin, Design of Business: Why Design Thinking is the Next Competitive Advantage, 2009, Harvard Business Press, Boston MA
5.	Tim Brown, Change by Design, 2009, Harper Collins
6.	Pavan Soni, Design your Thinking: The Mindsets,Tool sets and Skill Sets for Creative Problem-Solving, 2020, Penguin Books.

Links: NPTEL/ YouTube/ Faculty Video Link
https://www.youtube.com/watch?v=6_mHCOAAEI8
https://nptel.ac.in/courses/110106124
https://designthinking.ideo.com/
https://blog.experiencepoint.com/how-mcdonalds-evolved-with-design-thinking
https://www.coursera.org/lecture/uva-darden-design-thinking-innovation/the-ibm-story-iq0kE
https://www.coursera.org/lecture/uva-darden-design-thinking-innovation/the-meyouhealth-story-part-i- what-is-W6tTs
https://onlinecourses.nptel.ac.in/noc19_mg60/preview
https://nptel.ac.in/courses/109/104/109104109/
https://www.d-thinking.com/2021/07/01/how-to-use-storytelling-in-design-thinking/
https://www.worldofinsights.co/2020/10/infographic-8-design-thinking-skills-for-leadership-development/
https://www.youtube.com/watch?v=hFGVcx1Us5Y

Subject Name: Cloud Computing	**L-T-P [3-1-0]**
Subject Code: BMICA0401	**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: Adequate knowledge of Basics of Computer and computer network.	
Course Objective: To provide comprehensive knowledge of Cloud Computing concepts, technologies, and applications by introducing and researching state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations.	

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Define the fundamentals of cloud computing and computing techniques.	**K1**
CO2	Describe the concepts of virtualization and its role in cloud service delivery.	**K2**
CO3	Discuss various services and architecture of cloud	**K2**
CO4	Analyze the management of various cloud resources like instances, storage and network.	**K4**
CO5	Analyze the importance of cloud security solutions with monitoring and auditing.	**K4**

Syllabus

Unit No	Module Name	Topic covered	Pedagogy	Lecture Required (L+P)	Practical/ Assignment/ Lab Nos	CO Mapping

UNIT-1 Cloud computing and its infrastructure	**Introduction**	Introduction to Cloud Computing, Definition of Cloud, Evolution of Cloud Computing, Underlying Principles of Parallel and Distributed Computing, Cloud Characteristics, Scalability & Elasticity in Cloud, On-demand Provisioning, Multitenancy, Cloud economics.	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (1 to 4)	**CO1**
UNIT-2 Cloud virtualization basics	**Basics and need of Virtualization**	Basics and need of Virtualization, Types of Virtualizations, Implementation Levels of Virtualization, Virtualization Structures, Tools and Mechanisms, Virtualization of CPU, Memory- I/O Devices, VMM and its types, Virtual Machines, Virtualization tools, Virtualization Support and Disaster Recovery, Resource Provisioning and Resource Provisioning Methods	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (5-7)	**CO2**
UNIT-3 Service models and reference architecture	**Service Oriented Architecture**	Service Oriented Architecture, Systems of Systems, Web Services, REST, Publish Subscribe Model, Deployment Model- Public, Private and Hybrid Clouds, IaaS, PaaS, SaaS, Layered Cloud Architecture Design, Challenges and NIST Cloud Computing Reference Architecture, Benefits of CCRA, Architecture Overview- The conceptual Reference Model, Cloud Consumer, Cloud provider, Cloud Auditor, Cloud carrier, Scope of control between Provider and Consumer, IBM's Cloud Computing Reference Architecture (CCRA 2.0).	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (8)	**CO3**
UNIT-4 Resource management	**Resource Management**	Managed and Unmanaged resources in cloud, Instance Management, EC2, Azure Virtual Machine, Google Compute Engine. Storage Services: Block Storage, Elastic File Storage, Object Storage, S3, RDS, Dynamo DB, Backup, disaster recovery and storage migration. Network Services: VPC, Subnets, Routing, Security Groups, DNS, Direct Connect, VPC Endpoints.	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (9)	**CO4**

UNIT-5 Cloud security, monitoring and auditing	**Cloud Security, Monitoring and Auditing**	Challenges and Objectives; Cloud data life cycle; Common Attacks in Cloud; Security Standard: Confidentiality, Integrity, and Availability (CIA), Authentication and Authorization, Access controls: Role based access controls, multifactor authentication, Security policy management, IAM; Security Governance and Open Security Architecture; Monitoring and Auditing.	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (10)	**CO5**
Total				**40L+20P**		

Textbooks	
Sr No	**Book Details**
1.	Ritting house, John W., And James F. Ransome, ―Cloud Computing: Implementation, Management And Security, CRC Press, 2017.
2.	Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, "Distributed And Cloud Computing, From Parallel Processing To The Internet Of Things", Morgan Kaufmann Publishers, 2013.
3.	Raj kumar Buyya, Christian Vecchiola, S. Thamaraiselvi, ―Mastering Cloud Computing, Tata Mcgraw Hill, 2013.

Reference Books	
Sr No	**Book Details**
1.	Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing – A Practical Approach, Tata Mcgraw Hill, 2009.
2.	George Reese, "Cloud Application Architectures: Building Applications and Infrastructure In The Cloud: Transactional Systems For EC2 And Beyond (Theory In Practice), O'Reilly, 2009.

Links: NPTEL/ YouTube/ Faculty Video Link
https://www.youtube.com/watch?v=ynPjd9TKLc4
https://www.youtube.com/watch?v=b1b6JTYnbjU
https://www.youtube.com/watch?v=ab036IW3ASw
https://www.youtube.com/watch?v=HTb_VYOE4WM
https://www.youtube.com/watch?v=9CoMa2D7LwI

Subject Name: Database Management System					**L-T-P [3-1-0]**	
Subject Code: BMICA0402					**Applicable in Department: MCA Integrated**	
Pre-requisite of Subject: Students are expected to be familiar with Basics of Mathematics and Computer Science.						
Course Objective: To Introduce about database management systems, with an emphasis on how to organize, maintain and retrieve- efficiently, and effectively - information in relational & non-relational databases						
Course Outcomes (CO)						
Course outcome: After completion of this course students will be able to:						**Bloom's Knowledge Level(KL)**
CO1	Design ER and EER diagram of database for solving the real-world problems					**K2**
CO2	Apply and analyze the Structured Query Language (SQL) to solve the complex queries and implement normalization.					**K3, K4**
CO3	Implement the operators in complex queries and apply database connectivity for different applications					**K3**
CO4	Discuss PL/SQL and analyze transaction and concurrency control in transaction management.					**K2, K4**
CO5	Design and implement relational and non-relational database for the need of the real-world project.					**K3, K5**
Syllabus						
Unit No	**Module Name**	**Topic covered**		**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**
UNIT-1 Introduction of Database & Conceptual Designing	**Introduction about the DBMS**	Basic Concept: - Introduction of Data, Information, Database, DBMS, database system, structured, semi-structured and unstructured data. Database system Vs File system.			8L+4P	Assignment, Lab (1 to 8)	**CO1**
	Data models & Types of Data Models	Relational Database: Relation, Tuple, Attribute and Domain, Codd Rules, Design & Implement the ER Diagram					

	Data Modelling using the Entity Relationship Mode	ER model concepts, Degree of relationship, Notations for ER diagram, mapping constraints, reduction of ER diagrams to tables. Extended Entity Relationship Diagram & reduction of EER diagram to tables.	Lectures, PPTs, Notes			
	Introduction on SQ	Implements the DDL, DML, DCL & TCL: Introduction on SQL & Types of SQL commands:- DDL, DML, DCL, TCL.				
	Introduction on Relational Algebra & relational Calculus	Basic of Relation Algebra and Relational calculus.				
UNIT-2 Basic of SQL & Normalization ion	**Implementation of the Keys**	Keys & Types of Keys: - Super key, Candidate Key, Primary Key, Alternative Key, Composite Primary key, Foreign Key, unique and Composite Unique key.	Lectures, PPTs, Notes	8L+4P	Assignment, Lab (7 to 11)	**CO2**
	Implementation of Data Constrain	Data Constraint: - Null, Not Null, Default and check Constraint				
	Implementation of Aggregate function & clause	Use of Aggregate Function: -Min (), Max (), Count (), AVG (), Sum ().Use of Clause: Where, Group by, Having and Order by				
	Understand & Implement the normalization and different types of functions in SQL	Uses of String Functions in SQL, Uses of mathematical functions in SQL. Uses of Advanced Functions in SQL Armstrong's axioms. Functional Dependencies, Normalization & Types of Normalization, 1NF, 2NF, 3NF, BCNF. Multivalued Dependency, Join Dependency. Minimal Cover of FD's, Closure of an attribute, Lossless join decomposition.				
UNIT-3 Introduction of Complex Queries	**Operator & Predicates**	Operator & Predicates: - Like, Between, Aliases, Distinct, Limit		8L+4P	Assignment, Lab (12 to 16)	**CO3**
	Implementation of Logical operator	And, Or, Not				

	Set Theory Operator	Basic Set Operators: Selection, projection, rename, cross product, union, set difference	Lectures, PPTs, Notes			
	Derived Operators	Intersection, Division, Join. Inner Join: - Natural Join, Equi Join & Non Equi Join Outer Join:- Left Outer Join, Right Outer Join and Full Outer Join				
	Nested Query	Nested Query, Sub Query or Corelated Query: -IN, NOT IN, Exists, Not Exists, All, Any				
	Understand & Implementation the database connectivity	Database connectivity with Java or Python				
UNIT-4 Introduction of PL/SQL and Transaction & Concurrency control	**Implementation index, Views and Array**	Managing Indexes, Synonyms and Sequences, Managing Views, Managing Data in Different Time Zone	Lectures, PPTs, Notes	8L+4P	Assignment Lab (17 to 22)	**CO4**
	Implementation of PL/SQL	Introduction of PL/SQL, Implementation of PL/SQL Function Procedure, Trigger, Cursor				
	Implementation of Transaction management & concurrency control	Transaction system: - Life cycle of transaction, ACID Properties. Serial, non-serial schedule. Conflict Serializability. View Serializability. Recoverable Schedule, Cascade less schedule. Cascading rollback. Control Concurrency Techniques: Concurrency Control, Locking Techniques for concurrency control, 2-phase Locking protocol, strict 2 -phase locking protocol, rigorous 2-phase locking protocol.				
	Transaction & Data Control	Grant, Revoke, commit & Rollback				
UNIT-5 Introduction of NoSQL with MongoDB	**Understand NoSQL Concept and implement the**	Introduction of NoSQL Data Models: Document, Key Value, Column family, Graph. Overview of NoSQL Databases With their Types, Uses& Features of NoSQL Document Databases. CAP theorem, BASE Vs ACID.		8L+4P	Assignment, Lab (23 to 25)	**CO5**

	CURD operations		Lectures, PPTs, Notes			
	Introduction and Features of MongoDB	Sharding, Load Balancing, Indexing, Replication. MongoDB Shell & their commands Mongosh, MongoD, MongoDB Compass. MongoDB Collection, Document, Field &Value. MongoDB Operators, CRUD operations Implement the MongoDB Cursor				
	Relation and Aggregation in MongoDB	MongoDB Cursor & Methods, Relations in MongoDB, Aggregation in\MongoDB				
	Understand the concept of cloud database	Introduction of Cloud Database. MongoDB Cloud: - Atlas, Cloud Manager.				
Total				**40L+20 P**		

Textbooks

Sr No	Book Details
1.	Korth, Silbertz, Sudarshan," Database System Concepts", Seventh Edition, McGraw - Hill.(2019)
2.	Elmasri, Navathe, "Fundamentals of Database Systems", Seventh Edition Addison Wesley (2017)
3.	Ivan Bayross "SQL, PL/SQL The programming language of Oracle", Fourth Edition, BPB Publication. (2010)
4.	Brad Dayley "NoSQL with MongoDB in 24 Hours" Sams Publishing; 1st edition (2014)

Reference Books

Sr No	Book Details
1.	Thomas Cannolly and Carolyn Begg, "Database Systems: A Practical Approach to Design, Implementation and Management", Third Edition, Pearson Education, (2007)
2.	Raghu Ramakrishan and Johannes Gehrke "Database Management Systems" Third Edition, McGraw-Hill (2002)
3.	C J Date, "An Introduction to Database Systems", Eighth Edition , Pearson,. (2004)
4.	NoSQL and SQL Data Modeling: Bringing Together Data, Semantics, and Software First Edition by Ted Hills (2016)

Links: NPTEL/ YouTube/ Faculty Video Link
https://archive.nptel.ac.in/courses/106/106/106106220/
https://onlinecourses.nptel.ac.in/noc21_cs04/preview
https://nptel.ac.in/courses/106106093
https://swayam-uat-node1.appspot.com/proc_9i/preview
https://www.udemy.com/course/sql-to-nosql-database-handson-with-mongodb/

Subject Name: Software Engineering					**L-T-P [4-0-0]**	
Subject Code: BMICA0403					**Applicable in Department: MCA Integrated**	
Pre-requisite of Subject: Basic knowledge about software, along with any programming language.						
Course Objective: To enable students to develop methods and procedures for software development that can scale up for large systems and that can be used consistently to produce high, quality software at low cost and with a small cycle of time. Students will be able to understand the concepts of requirement engineering, designing and its principles, testing techniques and maintenance methods for effective software development.						
Course Outcomes (CO)						
Course outcome: After completion of this course students will be able to:						**Bloom's Knowledge Level(KL)**
CO1	Explain various software characteristics and analyze different software Development Models.					**K2**
CO2	Demonstrate the contents of an SRS and apply basic software quality assurance practices to ensure that design, development meet or exceed applicable standards.					**K3**
CO3	Discuss various methods for software design.					**K2**
CO4	Discuss testing strategy for software systems, employ techniques such as unit testing, Test driven development and functional testing.					**K2**
CO5	Explain maintenance and different techniques for project management.					**K2**
Syllabus						
Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**

UNIT-1 Introduction	**Introduction**	Introduction: Evolving role of Software, Software Characteristics, Software Crisis, Silver Bullet, Software Myths, Software Process, Software Engineering Phases, Team Software Process (TSP), Emergence of Software Engineering, Software process, Project, and Product. Software Process Models: SDLC, Waterfall Model, Prototype Model, Spiral, Model, Iterative Model, Incremental Model, V Process Model, Agile Methodology.	Lectures, PPTs, Notes	8L+4P	Assignment/ Experiment 1	**CO1**
UNIT-2 Software Requirement Specifications	**Software Requirement Specifications (SRS)**	Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modelling, Use Case Diagram, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS.	Lectures, PPTs, Notes	8L+4P	Assignment/Experiment (2,3)	**CO2**
UNIT-3 Software Design	**Software Design**	Software Design: Design principles, the design process; Design concepts: Abstraction, Refinement, Modularity (Cohesion and coupling), Software Architecture (Function Oriented Design, Object Oriented Design), Control Hierarchy (Top-Down and Bottom-Up Design), Structural partitioning, Data structure, Software procedure, Information hiding. Software Measurement and Metrics: Various Size Oriented Measures, Function Point, Design Heuristics for effective modularity, Cyclomatic Complexity Measures: Control Flow Graphs.	Lectures, PPTs, Notes	8L+4P	Assignment/Experiment (4-6)	**CO3**
UNIT-4 Software Testing	**Software Testing**	Software Testing: Testing Objectives, Unit Testing, Integration Testing, User Acceptance Testing, Regression Testing, testing for Functionality and Testing for Performance, Top Down and Bottom, Up Testing Strategies: Test Drivers and Test Stubs, Test Beds and Test Oracle, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies**:** Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards. Software Quality Assurance (SQA): Quality concepts, Software quality assurance, SQA activities, Formal approaches to SQA; Statistical software quality assurance; CMM, The ISO standard.	Lectures, PPTs, Notes	8L+4P	Assignment/Experiment (7-8)	**CO4**

UNIT-5 Project Maintenance and management concepts	**Software Maintenance**	Preventive, Corrective and Perfective Maintenance, Project Management concepts, Planning the Software Project, Cost of Maintenance, Estimation—Empirical Estimation COCOMO, A Heuristic Estimation Techniques, Staffing Level Estimation, Team structures, Risk analysis and management, Configuration Management, Software reengineering, Reverse Engineering, restructuring, Forward engineering, Clean Room software engineering, CASE Tools.	Lectures, PPTs, Notes	8L+4P	Assignment/Experiment (9-11)	**CO5**
Total				**40L+20P**		

Textbooks	
Sr No	**Book Details**
1.	KK Aggarwal and Yogesh Singh, "Software Engineering", New Age International Publishers, Fourth Edition,21 August 2022
2.	RS Pressman, Software Engineering: A Practitioners Approach, McGraw Hill ,9th edition 2020
3.	Rajib Mall, Fundamentals of Software Engineering, PHI Publication. ,5th Edition 2018

Reference Books	
Sr No	**Book Details**
1.	Pankaj Jalote, Software Engineering, Wiley.1 January 2010
2.	Kassem Saleh, "Software Engineering", Cengage Learning. 2009
3.	Ian Summerville, Software Engineering, Addison Wesley. 9th Edition. 29 October 2017

Links: NPTEL/ YouTube/ Faculty Video Link
https://youtu.be/x,jqSXYE4S4
https://youtu.be/mGkkZoFc,4I
https://youtu.be/sGxgZxwuHzc
https://youtu.be/BNk7vni,1Bo
https://youtu.be/8swQr0kckZI

Subject Name: Object Oriented Techniques Using Java	**L-T-P [0-0-8]**
Subject Code: BMICA0455	**Applicable in Department: MCA Integrated**
Pre-requisite of Subject: Understanding of basic Java syntax, familiarity with object-oriented concepts (classes, objects, inheritance, polymorphism), and problem-solving skills	
Course Objective: The objective of this course is to understand the object-oriented methodology, and its techniques to design stand alone and GUI applications using hands-on engaging activities.	

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Define the concepts of object-oriented programming and relationships among them needed in modeling.	**K1**
CO2	Demonstrate the Java programs using OOP principles and implement the concepts of lambda expressions.	**K3**
CO3	Analyze packages with different protection level resolving namespace collision and error handling concepts for uninterrupted execution of Java program	**K4**
CO4	Describe Concurrency control, I/O Streams and Annotations concepts by using Java program.	**K6**
CO5	Design and develop the GUI based application, Generics and Collections in Java programming to solve the real-world problem.	**K5**

Syllabus

Unit No	**Module Name**	**Topic covered**	**Pedagogy**	**Lecture Required (L+P)**	**Practical/ Assignment/ Lab Nos**	**CO Mapping**
UNIT 1 Basics of Java Programming	**Object Oriented Programming**	Introduction and Pillars of OOP with real life example, JVM architecture and its components.	Lectures, Code Walkthroughs, Hand-on Programming,	4L+10P	Assignment, Lab (1 to 13)	**CO1**

	Control Statements	Decision Making, Looping and Branching, Argument Passing Mechanism: Command Line Argument, Console Input	Problem Solving, Collaborative learning, competitive coding, Projects, Assessment			
	Class and Object	Object Oriented Concept in Java , Object Reference, Constructor, Abstraction: Abstract Class, Defining Methods, Garbage Collectionand finalize()Method etc				
UNIT-II OOPs features, arrays and lambda expressions	**Inheritance**	Introduction and Types of Inheritance in Java, Implementing Multiple Inheritance, Interface and its uses, Access Modifiers, Constructors, Use of "this" and "super" keyword	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment	4L+10P	Assignment, Lab (14 to 35)	**CO2**
	Polymorphism	Introduction and Types, Overloading and Overriding				
	Lambda expression	Introduction and Working withLambda Variables				
	Arrays	Introduction and its Types				
UNIT-III Packages, Exception Handling and String Handling	**Packages**	Introduction and Types, Access Protection in Packages, Importand Execution of Packages	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment	4L+10P	Assignment, Lab (36 to 55)	**CO3**
	Exception Handling	Introduction and Types, Exceptions vs. Errors, Handling of Exception Finally, Throws and Throw keyword, Multiple Catch Block, Nested Try and Finally Block				
	String Handling	Introduction and Types, Operations, Immutable String, Method of String class, String Buffer and String Builder class				
UNIT IV Concurrency in Java and I/O Stream	**Threads**	Introduction and Types, Creating Threads, Thread Life-Cycle, Thread Priorities, Daemon Thread, Runnable Class, Synchronizing Threads etc.	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative	4L+10P	Assignment, Lab (56 to 70)	**CO4**

	I/O Stream	Introduction and Types, Common I/O Stream Operations, Interaction withI/O Streams Classes	learning, competitive coding, Projects, Assessment			
	Annotations	Introduction, Custom Annotations and Applying Annotations				
UNIT-V GUI Programming, Generics and Collections	**GUI Programming**	Introduction and Types, Swing, Components and Containers ,Layout Managers and User- Defined Layout and Event Handling concept	Lectures, Code Walkthroughs, Hand-on Programming, Problem Solving, Collaborative learning, competitive coding, Projects, Assessment	4L+10P	Assignment, Lab (71 to 91)	**CO5**
	Generics	Introduction to Generic Classes, Initializing a Generic Object, Generic Cell Driver Class, Generic Methods, Use enumerated type				
	Collections	Introduction, Using Method References, Using Wrapper Class, Using Lists, Sets, Maps and Queues, Collection using Generics, Iterators				
Total				**20L+50P**		

Subject Name: Object Oriented Techniques using Java	**L-T-P [0-0-8]**

Course Objective: The objective of this course is to understand the object-oriented methodology, and its techniques to design stand alone and GUI applications using hands-on engaging activities.

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Define the concepts of object-oriented programming and relationship among them needed in modeling.	**K1**
CO2	Demonstrate the Java programs using OOP principles and implement the concepts of lambda expressions.	**K3**
CO3	Analyze packages with different protection level resolving namespace collision and implement the error handling concepts for uninterrupted execution of Java program.	**K4**
CO4	Describe Concurrency control, I/O Streams and Annotations concepts by using Java program	**K5**
CO5	Design and develop the GUI based application, Generics and Collections in Java programming language to solve the real-world problem.	**K5**

List of Practicals

Sr No	**Program Title**	**CO Mapping**
1	Write Programs Compile and run first Java file Byte Code and class file	**CO1**
2	Program to display default value of all Primitive data types	**CO1**
3	Implement the code using main() method to calculate and print the Totaland Average marks scored by a student from the input given through thecommand line arguments.	**CO1**
4	Assume that four command line arguments name, marks1, marks2, marks3 will be passed to the main() method in the below class with nameTotal And Avg Marks.	**CO1**
5	Write code which uses if-then-else statement to check if a given accountbalance is greater or lesser than the minimum balance. Write a class Balance Check with public method check Balance that takes one parameter balance of type double. Use if-then-else statement and print Balance is low if balance is less than 1000. Otherwise, print Sufficient	**CO1**

	balance.	
6	A class Number Palindrome with a public method is Number Palindrome that takes one parameter number of type int. Write a code to check whether the given number is palindrome or not. For example, Cmd Args : 333333 is a palindrome.	**CO1**
7	Write a class Fibonacci Series with a main method. The method receives one command line argument. Write a program to display Fibonacci seriesi.e. 0 1 1 2 3 5 8 13 21	**CO1**
8	Write a Java Program to find the Factorial of a given number.	**CO1**
9	Java Program to create a class, methods and invoke them inside main method.	**CO1**
10	Write a Java program to illustrate the abstract class concept. Create an abstract class Shape, which contains an empty method numberOfSides(). Define three classes named Trapezoid, Triangle and Hexagon extends theclass Shape, such that each one of the classes contains only the method numberOfSides(), that contains the number of sides in the given geometrical figure. Write a class Abstract Example with the main() method, declare an object to the class Shape, create instances of each class and call numberOfSides() methods of each class.	**CO1**
11	Java program to illustrate the static field in the class.	**CO1**
12	Java Program to illustrate static class.	**CO1**
13	Java program to explicit implementation of garbage collection by using finalize() method.	**CO1**
14	JAVA program to implement Single Inheritance.	**CO2**
15	JAVA program to implement multi-level Inheritance.	**CO2**
16	JAVA program to implement constructor and constructor overloading.	**CO2**
17	Write a Java program to access the class members using super keyword.	**CO2**
18	Java program to access the class members using this keyword.	**CO2**
19	Implement an interface named MountainParts that has a constant named TERRAIN that will store the String value “off_road”. The interface will define two methods that accept a String argument name newValue and two that will return the current value of an instance field. The methods are to be named: getSuspension, setSuspension, getType , setType.	**CO2**
20	Java program to demonstrate nested interface inside a interface.	**CO2**
21	Java program to demonstrate nested interface inside a class.	**CO2**

22	JAVA program implement method overloading.	**CO2**
23	JAVA program to implement method overriding.	**CO2**
24	Java program to implement lambda expression without parameter.	**CO2**
25	Java program to implement lambda expression with single parameter.	**CO2**
26	Java program to implement lambda expression with multi parameter.	**CO2**
27	Java program to implement lambda expression that iterate list of objects	**CO2**
28	Java program to define lambda expressions as method parameters.	**CO2**
29	Write a class CountOfTwoNumbers with a **public** method compareCountOf that takes three parameters one is arr of type int[] andother two are arg1 and arg2 are of type int and returns true if count of arg1 is greater than arg2 in arr. The return type of compareCountOf should be boolean. Assumptions: • arr is never null arg1 and arg2 may be same	**CO2**
30	JAVA program to show the multiplication of two matrices using arrays.	**CO2**
31	Java Program to search an element using Linear Search	**CO2**
32	Java program to search an element using Binary Search	**CO2**
33	Java Program to sort element using Insertion Sort	**CO2**
34	Java Program to sort element using Selection Sort – Largest element Method	**CO2**
35	Java program to Sort elements using Bubble Sort	**CO2**
36	Java program to create user defined package.	**CO3**
37	Java Program to create a sub- classing of package.	**CO3**
38	Implement the following: 1. Import package.*; 2. import package.classname; 3. Using fully qualified name.	**CO3**

39	Implement and demonstrate package names collision in Java.	**CO3**
40	Java program to handle and Arithmetic Exception Divided by zero	**CO3**
41	Java Program to implement User Defined Exception in Java	**CO3**
42	Java program to illustrate finally block	**CO3**
43	Java program to illustrate Multiple catch blocks	**CO3**
44	Java program for creation of illustrating throw	**CO3**
45	Java program to print the output by appending all the capital letters in theinput string.	**CO3**
46	Java program that prints the duplicate characters from the string with itscount.	**CO3**
47	Java program to check if two strings are anagrams of each other	**CO3**
48	Java Program to count the total number of characters in a string	**CO3**
49	Java Program to count the total number of punctuation characters existsin a String	**CO3**
50	Java Program to count the total number of vowels and consonants in astring	**CO3**
51	Java Program to show .equals method and == in Java	**CO3**
52	Given a string, return a new string made of n copies of the first 2 chars ofthe original string where n is the length of the string. The string may be any length. If there are fewer than 2 chars, use whatever is there. If inputis "Wipped" then output should be "WiWiWiWiWi".	**CO3**
53	Given two strings, a and b, create a bigger string made of the first char ofa, the first char of b, the second char of a, the second char of b, and so on.Any leftover chars go at the end of the result. If the inputs are "Hello" and "World", then the output is "HWeolrllod".	**CO3**
54	JAVA program to show the usage of string builder.	**CO3**
55	JAVA program to show the usage of string buffer.	**CO3**
56	Creating and Running a Thread	**CO4**
57	Implementing Runnable Interface	**CO4**
58	Synchronizing Threads with lock	**CO4**
59	Synchronizing Threads without lock	**CO4**

60	JAVA program to implement even and odd threads by using Thread class.	**CO4**
61	JAVA program to implement even and odd threads by using Runnableinterface.	**CO4**
62	JAVA program to synchronize the threads by using Synchronizestatements and Synchronize block.	**CO4**
63	Demonstrate the concept of type annotations in the JAVA programminglanguage.	**CO4**
64	JAVA program to implement that read a character stream from input fileand print it into output file.	**CO4**
65	Write a Java program that reads a text file and adds line numbers to eachline. The program should create a new file with the line numbers added to the beginning of each line.	**CO4**
66	JAVA program to implement that merge the content of two files (file1.txt,file2.txt) into file3.txt.	**CO4**
67	Write a Java program that reads two binary files and compares them byteby byte to determine if they are identical. Display a message indicating whether the files are the same or different.	**CO4**
68	Write a Java program that reads the contents of one file and copies themto another file.	**CO4**
69	Write a Java program that reads a text file and counts the number ofwords in it.	**CO4**
70	Write a Java program that reads a text file and counts the frequency ofeach word in it.	**CO4**
71	Program to create a frame with three button.	**CO5**
72	Program to display message with radio buttons.	**CO5**
73	Program to display "All The Best" in 5 different colors on screen.	**CO5**
74	Program to implement event handling in a button “OK”	**CO5**
75	Java Program to implement Border Layout	**CO5**
76	Java Program to implement Grid Layout	**CO5**
77	Java Program to implement Box Layout	**CO5**
78	Java Program to implement Card Layout	**CO5**
79	Java program to implement Generic class	**CO5**
80	Java program to illustrate Generic methods	**CO5**
81	Java program to implement wild card in generics	**CO5**

82	Java program to implement of methods of Hash Set	**CO5**
83	Java Program to implement methods available in HashMap class	**CO5**
84	Program to add, retrieve, and remove element from Array List	**CO5**
85	Create a method which can accept a collection of country names and add it to Array List with generic defined as String and return the List.	**CO5**
86	Create a method which can create a Hash Set containing values 1-10. The Set should be declared with the generic type Integer. The method should return the Set.	**CO5**
87	Develop a Java class with a method storeEvenNumbers(int N) using ArrayList to store even numbers from 2 to N, where N is a integer which is passed as a parameter to the method storeEvenNumbers(). The method should return the ArrayList (A1) created.	**CO5**
88	Create a method that accepts the names of five countries and loads them to an array list and returns the list.	**CO5**
89	Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the List.	**CO5**
90	Java program to implement autoboxing	**CO5**
91	Java program to implement unboxing	**CO5**

Required Software and Tools

1. Eclipse/Net beans

Textbooks

Sr No	Book Details
1.	"Java: The Complete Reference, 7th Edn", Herbert Schildt, McGraw Hill Education, 2017
2.	"Core Java: An Integrated Approach, New: Includes All Versions upto Java 8", R. Nageswara Rao and DT Editorial Services, Dreamtech Press, 2016
3.	"Programming with Java 7th Edition", E. Balagurusamy, McGraw Hill, 2023

Reference Books

Sr No	Book Details
1.	“Schaum's Outline Of Programming With Java / 2nd Edition”, Hubbard J.R., McGraw Hill, 2020
2.	“Programming In Java Revised 2Nd Edition”, Sachin Malhotra Saurabh Choudhary , Oxford University Press, 2018
3.	“Core Java Volume I - Fundamentals, 12th Edition”, Horstmann, Pearson Education, 2023
Links: NPTEL/ YouTube/ Faculty Video Link	
https://www.youtube.com/watch?v=AEo4KgwKYoU	
https://www.youtube.com/watch?v=5RkikYKPvpc&t=284s	
https://www.youtube.com/watch?v=bxcZ7cXbDI0&list=PLqleLpAMfxGAEfyXJyF-9UOs9C8dmir_Y	
https://www.youtube.com/watch?v=jmZfuI3lDK0	
https://www.youtube.com/watch?v=R0USRU90TOo https://www.youtube.com/watch?v=aXZrz8XKQpE https://www.youtube.com/watch?v=hKhlkx_6HeI&list=PLUDwpEzHYYLu9-xrx5ykNH8wmN1C1qClk	

Subject Name: Database Management System Lab	**L-T-P [0-0-4]**
Subject Code: BMICA0452	**Applicable in Department: MCA Integrated**

Course Objective: The objective of the course is to introduce about database management systems, with an emphasis on how to organize, maintain and retrieve -efficiently, and effectively - information in relational & non-relational databases.

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Design ER and EER diagram of database for solving the real-world problems	**K2**
CO2	Apply and analyze the Structured Query Language (SQL) to solve the complex queries and implement normalization	**K3, K4**
CO3	Implement the operators in complex queries and apply database connectivity for different applications.	**K3**
CO4	Discuss PL/SQL programs to solve complex problems in databases	**K2**
CO5	Design and implement relational and non-relational database for the need of the real-world project.	**K3, K5**

List of Practical's

Sr No	**Program Title**	**CO Mapping**
1	Creating ER Diagram for company Database. Company databases haveentities like employee, departments, projects and dependents also implement the relationship and cardinalities between the entities with their relevant attribute	**CO1**
2	Design an ER diagram for a travel agency that includes entities such as travelers, bookings, destinations, and itineraries. also implement the relationship and cardinalities between the entities with their relevantattribute	**CO1**
3	Convert Company ER Model to Relational Model(Represent entities and relationships in tabular form, represent attributes as columns, identifying keys).	**CO1**
4	Convert Travel Agency ER Model to Relational Model(Represent entities and relationships in tabular form, represent attributes as columns, identifying keys).	**CO1**
5	Data Definition Language Queries: Create Tables STUDENT, BOOK, TRANS.	**CO1**

	STUDENT(Rollno, Name, Branch, Year, Section, Hostel, F_name, Address) BOOK(Bookid, Title, Author, Publisher, Cost, Copies) TRANS(Rollno, Bookid, date_issue, date_return, fine) 1. Add a new attribute state in student table 2. Remove attribute address from student table 3. Modify the data type of state attribute 4. Change the name of attribute hostel to resident 5. Change a table's name, student to stud 6. Use truncate to delete the contents of trans table **7.** Remove the book table from database	
6	Data Manipulation Language Queries **1.** Insert at least 10 records in tables student, book and trans **2.** Show the contents in tables student, book and trans **3.** Find the name and branch of all students **4.** Find the name and rollno of all students who stay in hostel **5.** Find all distinct branches of students **6.** Delete the record of the student whose rollno is 204001 **7.** Delete all records of student table **8.** Delete all records of students whose section starts with capital A. **9.** Find the student names which have 'lk' in any position **10.** Find the student name where 'R' is in the second position **11.** Find the name of student whose name starts with 'V' and ends with 'A' **12.** Change the State of all students to 'BOMBAY' **13.** Change the state of student 'Vandana' to 'Goa' **14.** Apply arithmetic operators on cost column of book table for the book which has bookid = 1101	**CO1**
7	Queries with Constraints 1. Create the book table with Primary Key Constraint 2. Create trans table with foreign key Constraint 3. Create an Employee table with UNIQUE Constraint 4. Create Employee Table with Check Constraints 5. Create Supplier table with Default Constraint	**CO2**
8	Queries on TCL 1. Create Savepoints 2. Rollback to SavePoints 3. Use Commit to save on	**CO1**

9	Aggregate Functions: 1. Find the minimum, maximum, average and sum of costs of books 2. Count the total number of books present 3. Retrieve the average cost of all books authored by 'navathe'	**CO2**
10	String, Math andAdvanced Functions Implement The Following Functions: 1. ASCII() 2. CHAR_LENGTH() 3. CONCAT() 4. LCASE() 5. LOWER() 6. REPEAT() 7. REVERSE() 8. STRCMP() 9. ABS(X) 10. MOD(X,Y) 11. SIGN(X) 12. POWER(X,Y) 13. ROUND(X) 14. SQRT(X) 15. BIN() 16. COALESCE() 17. IF() 18. LTRIM 19. RTRIM 20. LPAD 21. RPAD 22. INITCAP	**CO2**
11	Queries on GROUP BY, HAVING AND ORDER BY Clauses 1. Display total costs of books by each author 2. Find the branch and the number of students in that branch for branches which have more than 1 student 3. Find all books sorted by title in ascending order and cost in descending order 4. Find the branch and the number of students in that branch	**CO2**
12	Queries on Operators 1. Find the title, author and cost of books which have cost equal to or greater than 200 and less than or equal to 600.	**CO3**

	2. Find the name , rollno and branch of students who are in 'CSE' branch or 'IT' branch 3. Find the title, author and cost of book for which cost is between 200 and 600 4. Find the title and author of book , which has the word 'NET' anywhere in its title. 5. Find the bookid and title of books with title either 'OS' or 'DBMS' 6. List the students who issued books on '1st may2000', '12 JAN 2021' ,'17 dec 2000' ,'10 Jan 2021' 7. Display all books which have cost more than the cost of all books authored by 'Yash' 8. Find all the distinct costs of books	
13	Join Operators 1. Perform Inner join on two tables 2. Perform Natural Join on two tables 3. Perform Left Outer Join on tables 4. Perform Right Outer join on tables 5. Perform Full Outer Join on tables	**CO3**
14	Set Theory Operators 1. Show the use of UNION operator with union compatibility 2. Show the use of intersect operator with union compatibility 3. Show the use of minus operator with union compatibility 4. Find the cartesian product of two tables	**CO3**
15	Queries on Set Theory Operators 1. List all books except 'Navathe' and 'Tannenbaum' in ascending order of costs 2. display all books that have not been issued so far 3. To display the students name who have been issued DBMS book by NAVATHE and OS book by TANNENBAUM. 4. To display the students name who have been issued DBMS book by NAVATHE OR OS book by TANNENBAUM	**CO3**
16	Complex Queries 1. display all books that have been issued so far 2. to display all the students of CSE IN year 2021 who are staying in the hostel. 3. To display students name who have issued OS book by Tannenbaum 4. To display the names of students who have not issued any book so far 5. To display the names of students who have issued at least one book so far. 6. To display students name along with the book issued to them 7. Find the names of students who have paid fine Rs1000 for the book 'OS in Depth'. 8. Retrieve the name of students who have issued the book which has the maximum cost. 9. Retrieve the names of students who have issued all books written by 'Korth'	**CO3**

17	Queries on Views 1. Create a view of student table 2. Find rollno and name from the created view where hostel is 'YES' 3. Create a view selecting rollno, name, branch, year and section from student table 4. Insert a row in the created view 5. Find all data in the created view 6. Update the created view by changing the name to 'GOGUL' for student whose rollno is 101 7. Compare the data of created view and the original table student 8. Delete the record of student whose rollno is '101' from the created view 9. Remove the view from database	**CO4**
18	Queries on Sequence, Index 1. Set the column to ROWNUM of a table 2. Create a sequence and set a column of a table to the created sequence. 3. Create an Index on the customer table	**CO4**
19	PL/SQL Programs 1. Write a PL/SQL Code to add two numbers 2. Write a PL/SQL code for Fibonacci series 3. Write a PL/SQL Code for greatest of 3 numbers 4. Write a PL/SQL code for area and circumference of a circle	**CO4**
20	PL/SQL Programs on Cursors 1. Write a Program using CURSOR to display ssn and salary of 1st record of employee 2. Write a program using cursors to display the ssn and salary of all employees and then print the count of employees	**CO4**
21	PL/SQL Programs on Triggers, Procedures and Functions 1. Write a Program using TRIGGER on UPDATE 2. Write a command to See the effect of trigger 3. Write a Program using PROCEDURE to increase the salary by Rs.1000 for Employee whose ssn is passed as an argument. 4. Write a procedure to update the address of an employee whose ssn and address are passed as arguments and the procedure returns the name of employee whose address is updated. 5. Write a function to return the total number of employees 6. Write a function to return the department number for which the department name is passed 7. Write a function to find the sum total of salaries of all employees. 8. Write a procedure to insert record in the department table 9. Write a code using EXCEPTION	**CO4**

22	PL/SQL Programs on Implicit Cursors 1. Insert a record using %ROWTYPE 2. Write a code using %NOTFOUND, %FOUND, %ROWCOUNT 3. Write a code using %TYPE	**CO4**
23	Mongo DB Queries 1. Create a collection. 2. Insert documents into Created Collection 3. Use insertMany() to insert more records 4. View the inserted records , raw and formatted 5. Select all documents in collection 6. Find count of all customers 7. .Show the records which have age equal to 18 8. Find all records which have fees between 2500 and 4500 9. retrieve all documents from the cust collection where status equals either "A" or "P" 10. Retrieve all documents where grade is equal to 'F' AND (fees is less than 3000 OR name starts with letter 'J') 11. Retrieve all documents where grade is equal to F OR fees is less than 4000 12. Update record with id 1, incrementing their fees by 50 13. Update the record of jack , set address to 'Delhi' and phoneno to '11221122' 14. Delete all records which have fees greater than 3000 15. Display only the grade and fees. 16. Get the grade, fees and custname of all records and sort by custname in ascending order. 17. Sort the Customers on their fees by descending order and get only first 2 records only 18. Update the postal code of 1st record and view it 19. select from the cust collection all documents where the grade equals "F": 20. Retrieve the document with exact value '5' 21. Retrieve documents where grade field contains values in given set. 22. Retrieve documents where grade field does not contain values in given set. 23. Retrieve all documents where grade is equal to "f" and fees is greater than or equal to 2000 and less than or equal to 4000. 24. Retrieve all documents which have grade not "F" nor "P". 25. Retrieve all documents where fees is not greater than or equal to 3000 26. Retrieve all documents where fees exists and is greater than or equal to 3000 27. Retrieve all documents which have fees type double 28. Retrieve all documents that have fees of type number 29. Retrieve all documents when we divide fees by 200 and remainder is 100	**CO5**

	30. Retrieve all documents that have regular expression ‘o’ in the custname field **31.** Retrieve all documents that have fees 2000 or 4300	
24	Connectivity with Database using Java or Python	**CO5**
25	Case Study Implementation of case Study on different domain 1. E-commerce Platform 2. Inventory Management 3. Railway System 4. Hospital Data Management 5. Voice-based Transport Enquiry System 6. SMS-based Remote Server Monitor system **7.** Banking System	**CO5**

Subject Name: Cloud Computing Lab		**L-T-P [0-0-4]**
Subject Code: BMICA0451		**Applicable in Department: MCA Integrated**
Course Objective: To provide hands-on experience in deploying, managing, and scaling applications on cloud platforms, emphasizing cloud services, virtualization, and distributed computing for efficient resource utilization		
Course Outcomes (CO)		
Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Discuss about the use AWS management console, create, and manipulate Amazon instances.	**K2**
CO2	Discuss the encrypting and controlling of S3.	**K2**
CO3	Describe how to create private and virtual private cloud.	**K2**
CO4	Create IAM group in cloud.	**K3**
CO5	Discuss the steps of Installation of Open Stack.	**K2**

List of Practical's

Sr No	Program Title	CO Mapping
1	Navigate the AWS Management Console.	**CO1**
2	Create and manipulate Elastic Compute Cloud instances.	**CO1**
3	Create AWS EC2 Virtual Machine Using AWS Console.	**CO1**
4	Monitoring Virtual Resources in AWS.	**CO1**
5	Getting Started with S3 in Cloud.	**CO2**
6	Working with EBS in AWS	**CO2**
7	Build a relational database server.	**CO2**

8	Create private cloud - Designing a Custom VPC (Virtual Private Cloud).	**CO3**
9	Create an IAM Group in Cloud.	**CO4**
10	Built a RESTful serverless API on AWS.	**CO5**
Required Software and Tools		
1. **AWS** 2. **Google Cloud**		

Subject Name: Project Based on Software Engineering Lab	**L-T-P [0-0-4]**
Subject Code: BMICA0459	**Applicable in Department: MCA Integrated**

Course Objective: To provide practical experience in software development processes, including requirements analysis, design, coding, testing, and maintenance, using modern tools and methodologies for building high-quality software systems.

Course Outcomes (CO)

Course outcome: After completion of this course students will be able to:		**Bloom's Knowledge Level(KL)**
CO1	Discuss ambiguities, inconsistencies, and incompleteness from requirement a requirements specification and state functional and non-functional.	**K3**
CO2	Explain different actors and use cases from a given problem statement and design use case diagram to associate use cases with different types of relationship	**K3, K4**
CO3	Design structural diagrams, behavioral diagram, interaction diagram for software	**K4**
CO4	Demonstrate the modern tools for testing	**K3**
CO5	Demonstrate the modern engineering tools for project management	**K5, K6**

List of Practical's

Sr No	**Program Title**	**CO Mapping**
1	Assignment of mini project based on real world problem and create the requirement statement for assigned mini project	**CO1**
2	Prepare an SRS document in line with the IEEE recommended standards on assigned project	**CO2**
3	Draw the use case diagram and DFD (0 level, 1 level, 2 level) for assigned project	**CO2**
4	Create Structural Diagram: Class diagram, Object diagram. Component diagram, Deployment diagram (assigned	**CO3**

	project)	
5	Create Behavioral diagram: Activity diagram, State diagram (assigned project)	**CO3**
6	Create Interaction diagram: Sequence diagram, Timing diagram, Collaboration diagram (assigned project)	**CO3**
7	Write Test cases for assigned project	**CO4**
8	Demo of JIRA software (Test case management & Agile software development).	**CO4**
9	Perform forward engineering using any language. (Model to code conversion)	**CO5**
10	Perform reverse engineering using any language. (Code to Model conversion)	**CO5**
11	Mini Project with CASE Tools	**CO5**
Required Software and Tools		
1. StarUML **2. DRAWio**		