ANILNEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (AUTONOMOUS)
ACCREDITEDBYNBA&NAACAFFILIATEDTOANDHRAUNIVERSITY

DEPARTMENTOFDEPT OF COMPUTER SCIENCE AND ENGINEERING

AcademicRegulations
Course Structure & Detailed Syllabus(R-23) Applicable for the batch admitted in 2023-24

II Year Course structure – CSE										
Semester –I										
CODE	**SUBJECT NAME**	**Category**	**Periods**				**Sessional Marks**	**Semester End Exam marks**	**Total Marks**	**Credits**
			L	**T**	**P**	**Total**				
23MA1103	Discrete Mathematics and Structure	BS	3	1	0	3	40	60	100	3
23CS4111	Data Structures and Algorithms	PC	3	1	0	3	40	60	100	3
23CS4112	Operating Systems	PC	3	1	0	3	40	60	100	3
23CS4113	Computer Networks	PC	3	1	0	3	40	60	100	3
23CS4114	Object Oriented Programing using JAVA	PC	3	1	0	3	40	60	100	3
23CS4211	Data Structures and Algorithms Lab	PC	0	0	3	3	50	50	100	1.5
23CS4212	Operating Systems Lab	PC	0	0	3	3	50	50	100	1.5
23CS4214	Object Oriented Programing using JAVA Lab	PC	0	0	3	3	50	50	100	1.5
23CS9211	UI /UX Design Tools	SC	0	0	2	2	50	50	100	1
23TP9101	Logical Reasoning and Corporate Skills	HS	0	0	2	2	50	50	100	1
23MC0103	Financial Literacy	MC	2	0	0	2	100	0	100	0
Total			**17**	**5**	**13**	**30**	**550**	**550**	**1100**	**21.5**

II Year Course structure – CSE										
Semester –II										
CODE	SUBJECT NAME	Category	Periods				Sessional Marks	Semester End Exam marks	Total Marks	Credits
			L	T	P	Total				
23MA1107	Probability and Statistics	BS	3	1	0	3	40	60	100	3
23CS4115	Formal Language and automata theory	PC	3	1	0	3	40	60	100	3
23CS4116	Database Management Systems	PC	3	1	0	3	40	60	100	3
23CS4117	Computer Organization and Microprocessor Interfacing	ES	3	1	0	3	40	60	100	3
23CS4118	Design and Analysis of Algorithms	PC	3	1	0	3	40	60	100	3
23CS4216	Database Management Systems Lab	PC	0	0	3	3	50	50	100	1.5
23CS4217	Computer Organization and Microprocessor Interfacing Lab	ES	0	0	3	3	50	50	100	1.5
23CS6211	Full Stack Development	JE	0	0	3	3	50	50	100	1.5
23CS9212	Mobile application Development	SC	0	0	2	2	50	50	100	1
23TP9102	Numerical Ability and Professional Communication Skills	HS	0	0	2	2	50	50	100	1
23MC0104	Entrepreneurship Development & IPR	MC	2	0	0	2	100	0	100	0
Total			**17**	**5**	**13**	**30**	**550**	**550**	**1100**	**21.5**

Semester –I

DISCRETE MATHEMATICAL STRUCTURES
(Common to CSE & IT)

23MA1103 **Credits:3**

Instruction : 3 periods & 1 Tutorial/Week Sessional Marks:40
End Exam : 3 Hours End Exam Marks:60

Prerequisites: Elementary knowledge of set theory, Matrices and functions.

Course Objectives:

This course will discuss fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science. Topics include logic, functions, relations, recurrence relations, fundamental concepts of number theory and graph theory.

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

1.	Identify logical skills in solving mathematical problems.
2.	Determine properties of binary relations, identify equivalence and partial order relations, and sketch relations.
3.	Analyze recurrence relations, generating functions and solving problems involving recurrence relations.
4.	Evaluate the concepts related divisibility, congruences and number theoretic functions and identify the structure of group, ring, and field.
5.	Explain the basic concepts of graph theory and develop a graph theoretical model for a real time situations.

CO-PO –PSO Mapping:

CO	PO												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2										1			
CO2	3	2										1			
CO3	3	2										1			
CO4	3	2										1			
CO5	3	2										1			

Correlation levels1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Mapping of Course Outcomes with Program Outcomes &Program Specific Outcomes:

CO-PO-PSO Justification	
1	CO1 deals with the logical inferences and first-order logic is used to make deductions and draw conclusions from given facts or premises.
2	CO2 deals with the relations are used to represent connections or relationships between objects in graph theory.
3	CO3 deals with the recurrence relations help identify recurring structures, repetitive patterns or sequences in problems, allowing for the formulation of efficient solutions.
4	CO4 deals with the concepts of prime numbers, modular arithmetic, and the Euclidean algorithm are used in cryptographic protocols to ensure secure communication, data encryption, and digital signatures and algebraic structures which are mainly used in encryption and decryption algorithms in computer science engineering.
5	CO5 deals with the concepts of the fundamental concepts of graph theory.

SYLLABUS

UNIT I **10 Periods**

MATHEMATICAL LOGIC

Fundamentals of logic – Logical inferences – Methods of proof of implication – First order logic and other proof methods – Rules of inference for quantified propositions – Pigeonhole principle – Mathematical induction.

UNIT II **10 Periods**

RELATIONS AND ALGEBRAIC SYSTEMS

Cartesian products of sets – Relations – Properties of binary relations in a set – Relation matrix and graph of a relation – Partition and covering of set – Equivalence relations – Composition of binary relations – Transitive closure of a relation – Partial ordering – Partially ordered set – Hasse diagram – Lattice.

UNIT III **10 Periods**

RECURRENCE RELATIONS

Generating functions of sequences – Calculating their coefficients – Recurrence relations – Solving recurrence relations – Method of characteristic roots – Non-homogeneous recurrence relations and their solutions.

UNIT IV **10 Periods**

NUMBER THEORY

Divisibility and Modular Arithmetic – Integer representations and algorithms – Primes and greatest common divisors – Solving congruences.

ALGEBRAIC STRUCTURES

Semi Groups – Monoids – Groups – Subgroups and their properties – Introduction to rings and fields. (Only definitions and examples)

UNIT V **10 Periods**

GRAPHS

Introduction to graphs – Types of graphs – Graphs basic terminology and special types of simple graphs – Representation of graphs and graph isomorphism – Euler paths and circuits – Hamilton paths and circuits – Planar graphs – Dual of a graph – Euler's formula – Graph coloring – Chromatic number.

TEXT BOOKS:

1. **Joe L. Mott, Abraham Kandel & T. P. Baker**, Discrete Mathematics for computer scientists &Mathematicians, Prentice Hall of India Ltd, New Delhi., 2008.
2. **Keneth. H. Rosen**, Discrete Mathematics and its Applications, 7/e, Tata McGraw-Hill, 2015.
3. **J. P. Tremblay, R. Manohar**, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill Publishing Company Limited,1997.

REFERENCE BOOKS:

1. **Richard Johnsonburg**, Discrete mathematics, 7/e, Pearson Education, 2008.
2. **Narsingh Deo**, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall of India, 2006.

DATA STRUCTURES & ALGORITHMS	
23CS4111	**Credits: 3**
Instruction: 3 Periods & 1 Tut/Week	Sessional Marks: 40
End Exam: 3 Hours	End Exam Marks: 60

Prerequisites:

Basic Knowledge of Programming Fundamentals and Problem Solving

Course Objectives:

The course should enable the students:

- To acquire knowledge on several linear and nonlinear data structures like stacks, queues, linked list, trees and graphs.
- To have better insight into to learn various sorting and searching techniques.
- To exercise the applications of data structures.
- To have a good understanding of problem-solving using data structure tools and techniques.

Course Outcomes:

Students will be able to.

1. Analyze the complexities of recursive and non-recursive algorithms and implement linear, binary, interpolation, hashing searching techniques and sorting techniques namely bubble, insertion, selection, quick, merge sort.
2. Apply ADT concepts such as stacks and queues for solving infix to postfix, postfix evaluation and queue applications.
3. Apply the concepts of dynamic memory allocation to implement Linked Lists.
4. Design and implement the Nonlinear data structures (trees) to optimize the solution.
5. Design and Implement Warshall's Algorithm, Shortest path Algorithm-Dijkstra's Algorithm, Minimum cost spanning trees (Prims and Kruskal's algorithms), Graph traversals (Breadth first search and Depth first Search algorithms.)

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	3	3	2	3	-	-	-	-	-	-	-	1	3	-
	2	2	2	3	2	-	-	-	-	-	-	-	1	2	-
	3	2	2	3	2	-	-	-	-	-	-	-	1	3	-
	4	2	2	3	2	-	-	-	-	-	-	-	1	2	-
	5	2	3	3	3	-	-	-	-	-	-	-	1	3	-

SYLLABUS

UNIT-I: 12 periods

Introduction: Data Structure operations, Fundamentals of analysis of algorithms and efficiency – Asymptotic Notations and Basic Efficiency classes.

Searching &Sorting: Sequential search, binary search, Interpolation Search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Techniques-Open hashing, Closed hashing. Insertion Sort, Shell sort, Quick Sort, Merge Sort.

Learning Outcomes:

1. Analyze the complexity of Algorithms.
2. Implement the searching and sorting algorithms.

UNIT-II: 12 periods

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Applications of stack: Conversion of Infix to prefix and Postfix Expressions, Evaluation of Postfix & Prefix expressions using stack, Recursion, Towers of Hanoi Problem.

Queues: Array representation and implementation of queues, Operations on Queue: Insert, Delete, Full and Empty. Circular queue, Applications of Queues.

Learning Outcomes:

1. Apply ADT to implement Stack and queue
2. Apply ADT to implement applications of stack and queue.

UNIT-III: 12 periods

Linked list: Representation and Implementation of Singly Linked Lists, Traversing and Searching of Linked List, Insertion and deletion to/from Linked Lists, doubly linked list, Circular doubly linked list, implementing priority queue using Linked List, Polynomial Representation using Linked list & addition.

Learning Outcomes:

1. Implement singly linked list, Doubly Linked List, Circular doubly linked list and applications.
2. Develop the skills to implement data structures and algorithms using linked lists, such as priority queues and polynomial representations.

UNIT-IV: 12 periods

Trees: Basic terminology, Binary Trees-Full Binary Tree, Complete Binary Tree, Extended Binary Tree, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees. Binary Search Tree (BST), Insertion and Deletion in BST, AVL Trees-Rotations in AVL trees, Insertion and Deletion in AVL.

Learning Outcomes:

1. Design and implement BST along with its operations.
2. Implement AVL tree along with its operations.

UNIT-V: **12 periods**

Graphs: Graphs Terminology, Types of Graphs, Representations- Adjacency Matrices, Adjacency List, Path or Transitive Closure of a Graph, Warshall's Algorithm, Graph Traversals, Shortest path Algorithm- Dijkstra's Algorithm, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

Learning Outcomes:

1. Implement Graph Traversals algorithm and Minimum Cost Spanning Trees algorithms.
2. Implement Warshall's Algorithm, Shortest path Algorithm-Dijkstra's Algorithm.

TEXT BOOKS

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education, 2nd Edition, 1996

REFERENCE BOOKS

1. E.Horowitz and Sahani, "Fundamentals of Data Structures", W H Freeman& Co Publication, 1983.
2. S. Lipschutz, "Data Structures", McGraw Hill Publications, 1986.
3. P. Dey & M. Ghosh, "Programming in C" , Oxford Univ. Press, 2012
4. ISRD Group, "Data Structures through C++", McGraw Hill, 2011.

WEB RESOURCES:

1. https://nptel.ac.in/courses/106/102/106102064/
2. https://www.coursera.org/learn/data-structures?action=enroll&specialization=data-structures-algorithms
3. https://www.udacity.com/course/data-structures-and-algorithms-nanodegree--nd256

OPERATING SYSTEMS	
23CS4112	**Credits : 3**
Instruction: 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

- Basic programming knowledge.

Course Objectives:

- Describe the main components of an operating system and their functions.
- To gain the knowledge on Process Scheduling
- Learning process synchronization and deadlock conditions
- Apply paging concept in memory management
- Illustrate File handling in secondary storage.

Course Outcomes:

CO	At the end of the course the student will be able to
1.	Explore the structure of the operating system, functionality and services provided by the operating system and interpret the basic concept of shell programming.
2.	Implement the CPU Scheduling algorithms (Pre-emptive and Non Pre-emptive)
3.	Determine the concept of Process synchronization and the concept of resource allocation and deadlock.
4.	Analyze various memory management mechanisms for contiguous and non-contiguous memory.
5.	Comprehend the structure of the file system in secondary storage structures.

Mapping of Course Outcomes with Program Outcomes:

CO	PO												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
1	1	1	1	-	-	-	-	-	-	-	-	-	2	-
2	2	2	3	1	-	-	-	-	-	-	-	-	2	-
3	2	2	3	1	-	-	-	-	-	-	-	-	2	-
4	2	2	3	1	-	-	-	-	-	-	-	-	2	-
5	2	2	3	1	-	-	-	-	-	-	-	-	2	-

Syllabus

UNIT I **12 Periods**

Introduction to OS: Overview, Operating system Functionalities, process management, memory management, storage management, Types of Operating system, operating system structures, system calls, Types of system calls, system programs.
Introduction to Shell Programming: Commands and Shell script.

Learning outcomes: At the end of this Unit, Students will be able to

1. Define the responsibilities of an operating system
2. Implement the basic shell programs.

UNIT II **12 Periods**

Processes: Process concept, Process scheduling, Operations on processes, Inter process communication, Communication in client-server systems.
Threads: Overview, Multithreading models.
CPU Scheduling: Scheduling criteria, Scheduling algorithms, Algorithm Evaluation.

Learning outcomes: At the end of this Unit, Students will be able to

1. Demonstrate the different modes of communication among processes and multi-threading models.
2. Analyze the CPU scheduling algorithms and their performance evaluation.

UNIT III **12Periods**

Process Synchronization: The critical-section problem, Peterson's solution, Synchronization hardware, Mutex Locks, Semaphores, Classic problems of synchronization, Monitors.**Deadlock:** System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

Learning outcomes: At the end of this Unit, Students will be able to

1. Implement the different solutions for process synchronization.
2. Define the concept of deadlock and Identify the different ways to handle deadlock like prevention, detection, avoiding and recovery.

UNIT IV **12Periods**

Memory Management: Background, Swapping, Contiguous memory allocation, Segmentation, Paging, Structure of the page table.
Virtual Memory: Background, Demand paging, Page replacement Algorithms, Allocation of frames, Thrashing.

Learning outcomes: At the end of this unit, students are will be to

1. Distinguish between contiguous and non-contiguous memory allocation methods in memory management.
2. Demonstrate the concept of virtual memory management.

UNIT V **12Periods**

File Systems Interface: File concept, Access methods, Directory structure, File system structure, File system implementation, Directory implementation, Allocation methods, Free-space management.

Secondary Storage Structure: Mass storage structures, Disk structure, Disk attachment, Disk scheduling, Disk management, Swap space management.

Learning outcomes: At the end of this unit, students are will be to

1. Demonstrate the concept of file system, various file access methods and Protection in files.
2. Demonstrate the concept of mass storage structures and Analyze the various disk scheduling algorithms

TEXT BOOKS

1. Silberschatz, Galvin and Gagne, "Operating System Principles", 9th Edition, Wiley India Pvt Ltd, 2015.
2. Sumitabha Das, "Unix Concepts and Applications", 4th Edition. TMH, 2006.
3. Yashwanth Kanitkar, "Unix Shell programming", 1st Edition, BPB Publisher, 2010.

REFERENCES

1. Andrew S. Tanenbaum, "Modern Operating Systems", 4th Edition, Pearson Education, 2015.
2. William Stalling, "Operating Systems: Internals and Design Principles", 9th edition, PHI, 2018.

WEB REFERENCES:

1. https://nptel.ac.in/courses/106/106/106106144/
2. https://nptel.ac.in/content/storage2/courses/106108101/pdf/PPTs/Mod13.

COMPUTER NETWORKS	
23CS4113	**Credits : 3**
Instruction : 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:
Basic knowledge of Computer fundamentals.

Course Objectives:

- To educate concepts ,vocabulary and techniques currently used in the area of Data communication**s**
- To interpret the Digital encoding Techniques in Data Communication
- The students should understand the layers of networking devices.
- To gain knowledge in different mediums used for data transfer.
- They should be familiar with a few networking protocols.
- They should study the different types of networks and topologies of networks.

Course Outcomes: By the end of the course, the student will be able to

CO1	Illustrate the basic data communications model, comparing OSI and TCP/IP models.
CO2	Acquire the knowledge of transmission media and encoding techniques.
CO3	Analyze the error and flow control mechanisms.
CO4	Acquire the knowledge of network protocols.
CO5	Acquire the knowledge of transport and application layer protocols.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	3	2	-	1	3	-	-	-	2	1	-	3	1	2
	2	3	2	2	2	1	-	-	-	2	2	-	2	1	2
	3	3	2	2	2	1	-	-	-	2	2	-	2	1	2
	4	3	2	2	2	1	-	-	-	2	1	-	2	1	2
	5	3	2	2	2	1	-	-	-	2	2	-	3	1	2

SYLLABUS

UNIT I **12Periods**

Introduction: Data Communications**Internetworking:** A CommunicationModel, Protocol Architecture. **Evolution of computer networks**: Types of Networks, Topologies
Network Architecture: OSI, TCP/IP models.

Learning outcomes: At the end of this Unit, Students are able to

1. Explain the Representations used for defining data communications with the state of art.
2. Analyze the performance comparison between ISO-OSI model and TCP/IP models.

UNITII **12Periods**

Guided Transmission Media, Wireless Transmission, Types of Signals, Encoding Techniques, Switching Techniques, Multiplexing -Types.

Learning outcomes: At the end of this Unit, Students are able to

1. Analyze the signals with encoding techniques to present the target data.
2. Learn switching methods

UNIT III **12Periods**

Framing, Error detection and correction, HDLC, Flow control, medium access control, Token Ring(802.3), Wireless LAN(802.11).

Learning outcomes: At the end of this Unit, Students are able to

1.Analyze error and flow control mechanisms.
2.Analyze access control and Types of LANs.

UNIT IV **12Periods**

Internet addressing, IP (v4 and v6), ARP, ICMP, CIDR, **Routing algorithms**: Shortest path first, Distance Vector, Link State Routing algorithms (RIP, OSPF, BGP),Tunneling.

Learning outcomes: At the end of this Unit, Students are able to

1. Analyze the network layer protocols with addressing
2. Compare Routing algorithms

UNIT V **12Periods**

UDP, TCP, congestion control, Introduction to Quality of Service, WWW, HTTP/HTTPS, SMTP,MIME,DNS, SNMP.

Learning outcomes: At the end of this unit, students are able to

1. Explore the usage of various application layer protocols.
2. Understanding of the roles and functionalities of transport layer protocols.

TEXT BOOKS:

1. William Stallings, "Data and Computer Communications", 8th Edition, Pearson Education Inc., 2010.
2. A. S.Tanenbaum and D.J. Wetherall, Computer Networks, Fifth Edition, Pearson, 2011.

REFERENCES

1. Behrouz A Forouzan “Data Communications and Networking”, 5th Edition, Tata McGraw-Hill, 2012

WEB REFERENCES:

1.https://memberfiles.freewebs.com/00/88/103568800/documents/Data.And.Computer.Communications.8e.WilliamStallings.pdf
2.https://nptel.ac.in/courses/106/106/106106144/
4.https://en.wikipedia.org/wiki/Xv6.
5.https://nptel.ac.in/content/storage2/courses/106108101/pdf/PPTs/Mod_13.pdf

OBJECT ORIENTED PROGRAMMING USING JAVA	
23CS4114	**Credits : 3**
Instruction : 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Pre requisites:

- Basic knowledge of computer fundamentals.
- Student must have knowledge of some programming languages (such as C, C++)

Course Objectives:

- To gain knowledge of object-oriented programming concepts and apply them in problem solving.
- To learn the basics of Java console and GUI based programming.

Course Outcomes:

At the end of the course the student will be able to

CO1	Analyze the basic concepts of object oriented programming.
CO2	Create classes and objects for developing the applications.
CO3	Create packages for different methods and handle exceptions.
CO4	Implement multithreading techniques and file handling concepts.
CO5	Develop GUI based applications.

CO-PO MAPPING:

CO	**PO**												**PSO**	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
1	2	2	3	2	2	-	-	-	-	-	2	2	2	-
2	2	3	3	2	2	-	-	-	-	-	2	2	2	-
3	2	3	3	2	2	-	-	-	-	-	2	2	2	-
4	2	2	3	2	2	-	-	-	-	-	2	2	2	-
5	2	2	3	2	2	-	-	-	-	-	2	2	2	-

Correlation Levels 1 2 3 Defined as Below

1 **High: Strong Correlation**
2 **Medium: Moderate Correlation**
3 **Low: Slight**

SYLLABUS

UNIT-I **12 periods**

Introduction: OOPS features, Applications, Simple programson java,Data types, variables, operators, console input and output, Control structures, arrays, Strings**.**

Learning Outcomes: At the end of this unit the students will be able to

1. Identify the object and understand object oriented principles
2. To gain the fundamentals of java programming to handle string operations

UNIT – II **12 periods**

Classes and Objects: Introduction to Classes, objects, constructors - types, overloading. Method- overloading. Keywords – this, static, final. Garbage Collection.

Inheritance – Types, Method overriding,super keyword, interface, abstract class.

Learning Outcomes: At the end of this unit the students will be able to

1. Identify the object and understand object oriented principles
2. Create class, constructor and can handle string operations

UNIT –III **12 periods**

Packages - Defining, creating and accessing a package, importing packages, member access rules

Exception handling – Fundamentals, Exception types, use of try and catch, throw, throws, finally, built-in exceptions, user defined exceptions.

Learning Outcomes: At the end of this unit the students will be able to

1. Derive a class from existing class or from interface
2. Define a package and importing class from package
3. Handle predefined Exceptions and can define custom exceptions

UNIT-IV **10 periods**

Multithreading – Thread life cycle, Thread implementation,Thread Priorities, synchronization, producer-consumer problem.

File I/O–FileInputStream, FileOutputStream, FileReader, FileWriter, equenceInputStream.

Learning Outcomes: At the end of this unit the students will be able to

1. To acquire knowledge and split a complex task into multiple threads and the Multiple Thread operations
2. Learn the streams in java

UNIT-V **12 periods**

AWT and Swings

Components, Swing components, Panel, Menus, Layout Managers.

Event Handling: The Delegation event model, Event classes, Event Listeners, Adapter Classes. Handling events using Action, Mouse, Key and Window events.

Learning Outcomes: At the end of this unit the students will be able to

1. Design Swing Applet class with html tag
2. Design GUI components using AWT.
3. Define Event Handling on the components using Delegation event model.

TEXT BOOKS

1. Herbert Schildt, “JAVA The Complete Reference”, TataMcGrawHill,seventh edition.
2. E Balagurusamy, “Programming with JAVA - A Primer” – Third Edition.

REFERENCES BOOKS

1. P.J. Deitel and H.M. Deitel, “Java for Programmers”, Pearson education (OR) P.J. Deitel and H.M. Deitel, “Java: How to Program”, PHI.
2. P. Radha Krishna, “Object Oriented Programming through Java”, Universities Press.
3. Bruce Eckel, “Thinking in Java”, Pearson Education
4. Bruce Eckel, “Programming in Java”, Pearson Education
5. S. Malhotra and S. Choudhary, “Programming in Java”, Oxford Univ. Press.

WEB RESOURCES:

1. **https://www.decodejava.com**
2. **http://java.candidjava.com/tutorial/Core-Java-tutorial-with-example.html**
3. **https://docs.oracle.com/javase/tutorial/java/nutsandbolts**
4. **https://developer.ibm.com/tutorials/j-introtojava1**
5. **https://codesjava.com/programing-language-overview**

DATA STRUCTURES AND ALGORITHMS LAB	
23CS4211	**Credits : 1.5**
Instruction **:** 3 Periods / week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites:

- Basic knowledge about problem solving

Course objectives:

- The course is designed to develop skills and analyze simple linear and non- linear data structures.
- It strengthens the ability of the students to identify and apply the suitable data structure for the given real-world problem.
- It enables them to gain knowledge in practical applications of data structures.

Course Outcomes:

Student will be able to:

1. Implement the techniques for searching and sorting (quick and merge).
2. Implement stack and queue data structures and their applications.
3. Implement Linked list data structures and their applications.
4. Implement operations on trees and graphs data structures (Create, search and traversal mechanisms)

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	2	2	2	1	-	-	-	-	-	-	-	1	1	-
	2	1	2	2	1	-	-	-	-	-	-	-	1	1	-
	3	2	2	2	1	-	-	-	-	-	-	-	1	2	1
	4	2	2	2	1	-	-	-	-	-	-	-	1	2	-

LIST OF PROGRAMS

1. Write a program to sort the given array of N elements using divide and conquer method (merge sort and quick sort algorithms) **CO1**
   Constraints: 1<N<1000
   Sample Input array: 87, 36, 9, 12, 24, 5, 78, 567, 456, 34, 96, 45, 39, and 89,123
   Sample Output array: 5, 9, 12, 24, 34, 36, 39, 45, 78, 87, 89, 96, 123, 456, and 567

2. Write a Program to search whether an item K present in an array of N elements (Using Linear and binary Search algorithms) **CO1**
   Constraints: 1<K<1000
   1<N<1000
   Sample Input array: 45, 78,123, 48, 34, 89, 67, 54, and 74,543
   Search Item: 34 Search Item: 343
   Output: Key Found Output: Key Not Found

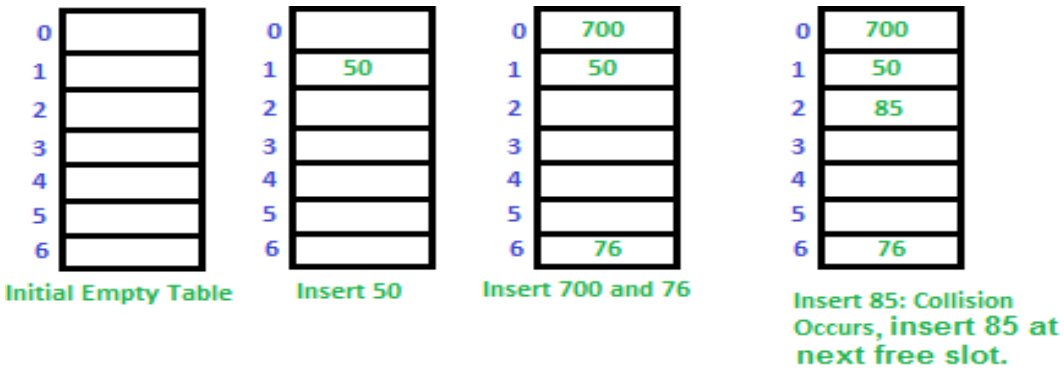
3. Write a program to store k keys into an array of size n at the location computed using a hash function, loc = key % n, where k<=n and k takes valuesfrom
   [1 to m], m>n. **CO1**

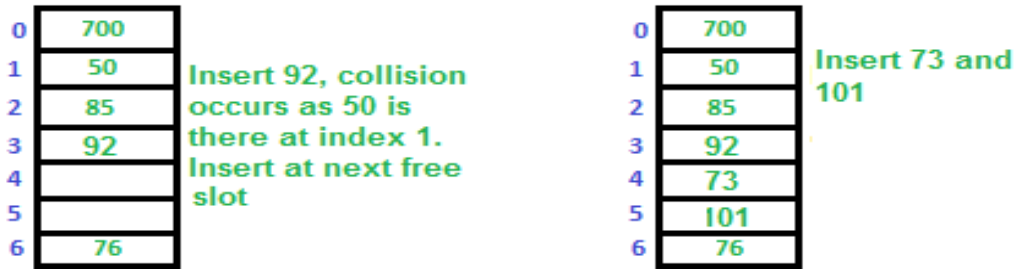
4. Design, Develop and Implement a program to handle the collisions using the following collision resolution Technique **CO1**

   a) Linear probing: In linear probing, we linearly probe for next slot, let store k keys into an array of size S at the location computed using a hash function, hash(x) where k<=n and k takes values from [1 to m], m>n.

   Constraints: If slot hash(x) % S is full, then we try (hash(x) + 1) % S If (hash(x) + 1) % S is also full, then we try (hash(x) + 2) % S
   If (hash(x) + 2) % S is also full, then we try (hash(x) + 3) % S
   .....
   .....

   Sample Test Case:
   Let us consider a simple hash function as "key mod 7" and sequence of keys as 50, 700, 76, 85, 92, 73, 101.

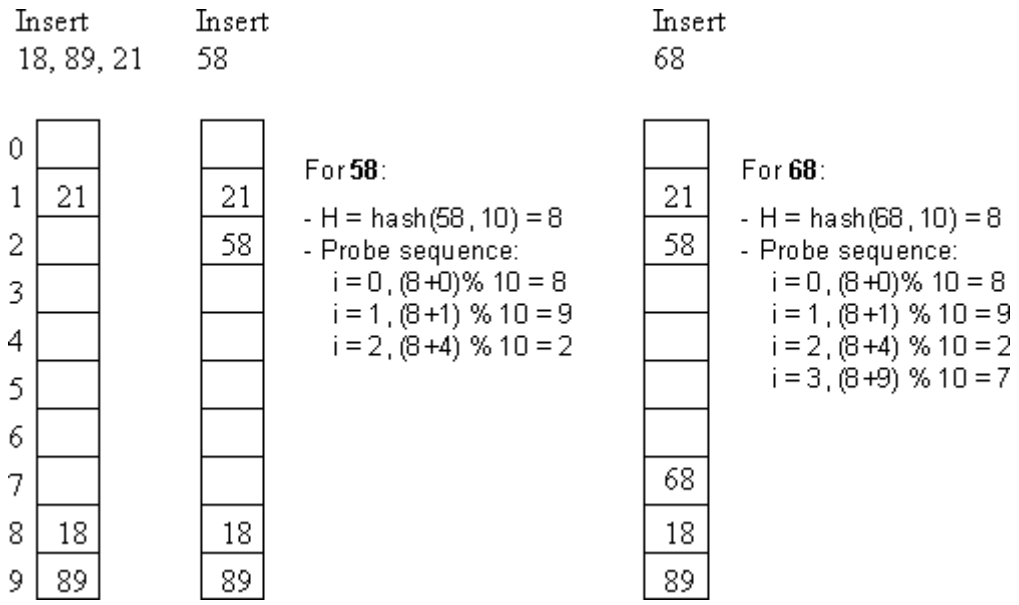




Quadratic probing: Quadratic Probing we look for i2‘th slot in i’th iteration, let store k keys into an array of size S at the location computed using a hash function, hash(x) where k<=n and k takes values from [1 to m], m>n.

Constraints: let hash(x) be the slot index computed using hash function. If slot hash(x) % S is full, then we try (hash(x) + 1*1) % S
If (hash(x) + 1*1) % S is also full, then we try (hash(x) + 2*2) % S If (hash(x) + 2*2) % S is also full, then we try (hash(x) + 3*3) % S
.....

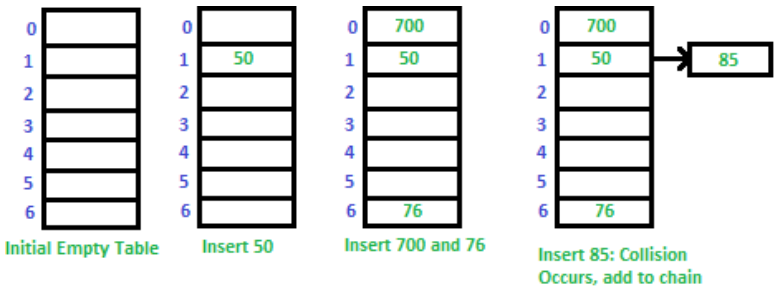
Sample Test Case:

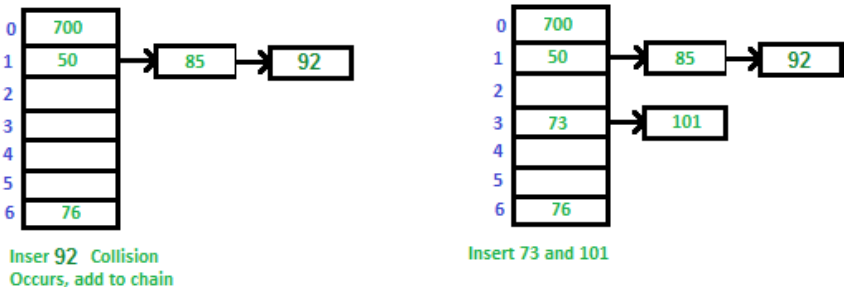


b) Separate Chaining: The idea is to make each cell of hash table points to a linked list of records that have same hash function value.
Let us store K keys into hash table of size S, where k<=n and k takes values from [1 to m], m>n.

Sample Test Case:

Let us consider a simple hash function as “key mod 7” and sequence of keys as 50, 700, 76, 85, 92, 73, 1





5. Design, Develop and Implement a menu driven Program for the following.

a) Operations on **STACK** of Integers (Array Implementation of Stack with maximum size **MAX**) **CO2**

1. **Push** an Element on to Stack
2. **Pop** an Element from Stack
3. Demonstrate **Overflow** and **Underflow** situations on Stack
4. Display the status of Stack
5. Exit

b) Operations on **QUEUE** of Characters (Array Implementation of Queue with maximum size **MAX**)

1. Insert an Element on to QUEUE
2. Delete an Element from QUEUE
3. Demonstrate **Overflow** and **Underflow** situations on QUEUE
4. Display the status of QUEUE
5. Exit

Note: Support the program with appropriate functions for each of the above operations

6. Design, Develop and Implement a Program for the following. **CO2**

a) Converting an **Infix Expression to Postfix Expression**. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^(Power) and alphanumeric operands.

b) Evaluation of postfix expression with single digit operands and operators: +,-, *, /, %, ^

7. Design, Develop and Implement a menu driven Program for **Circular Queue**: **CO2**

1. Insert an Element on to Circular QUEUE
2. Delete an Element from Circular QUEUE
3. Demonstrate *Overflow* and *Underflow* situations on Circular QUEUE
4. Display the status of Circular QUEUE
5. Exit

Support the program with appropriate functions for each of the above operations

8. Design, Develop and Implement a program to do the following using a **single linked list.** **CO3**

a) Stack- In single linked list store the information in the form of nodes. Create nodes using dynamic memory allocation method. All the single linked list operations perform based on Stack operations LIFO (last in first out).

A stack contains a top pointer. Which is “head” of the stack where pushing and popping items happens at the head of the list. first node has null in link field and second node link have first node address in link field and so on and last node address in “top” pointer.

Stack Operations:

1. push() : Insert the element into linked list nothing but which is the top node of Stack.
2. pop() : Return top element from the Stack and move the top pointer to the second node of linked list or Stack.
3. peek(): Return the top element.
4. display(): Print all element of Stack.

b) Queue- All the single linked list operations perform based on queue operations FIFO (First in first out).

In a Queue data structure, we maintain two pointers, *front* and *rear*. The *front* points the first item of queue and *rear* points to last item.

1. enQueue() This operation adds a new node after *rear* and moves *rear* to the next node.
2. deQueue() This operation removes the front node and moves *front* to the next node.
3. Display() Display all elements of the queue.

9. Design, Develop and Implement program to do the following operations using a **double linked list.** **CO3**

1. Create a linked list
2. Insert a node at the beginning
3. Insert a node at the end
4. Insert a node after given node
5. Insert a node before given node
6. Delete first node
7. Delete last node
8. Delete in between node
9. Search given element

10. Design, Develop and Implement a menu driven Program for the following operations on**Binary Search Tree (BST)** of Integers **CO4**

a. Create a BST of N Integers: 13, 3, 4, 12, 14, 10, 5, 1, 8, 2, 7, 9, 11, 6, 18
b. Traverse the BST (either inorder, preorder or postorder)
c. Search the BST for a given element (KEY) and report the appropriate message
d. Exit

11. Design, Develop and Implement a Program for the following operations on Graph(G) of Cities. **CO4**

1. Create a Graph of N cities using Adjacency Matrix.
2. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method

12. Design, Develop and Implement a Program to find the Minimum Spanning Tree (MST) using the following. **CO4**

1. Prim's Algorithm
2. Kruskals Algorithm

OPERATING SYSTEM LAB	
23CS4212	**Credits : 1.5**
Instruction : 3 Periods / week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites: Basic programming Knowledge

Course objectives:

- Implement the basic shell script, UNIX commands and system calls.
- Examine the process, memory and file management.
- Solve the problems related to process synchronization.

Course Outcomes:

At the end of the course the student will be able to

CO1	Implement the Unix Shell programming on the given system configuration.
CO2	Implement process creation and communication between processors.
CO3	Demonstrate the process scheduling, process synchronization, deadlock avoidance.
CO4	Apply memory management techniques and file allocation methods.

Mapping of Course Outcomes with Program Outcomes:

CO	PO												PSO	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
1	1	2	2	-	-	-	-	-	-	-	-	-	-	-
2	1	2	2	-	-	-	-	-	-	-	-	-	2	-
3	2	2	3	1	-	-	-	-	-	-	-	-	2	-
4	2	2	3	1	-	-	-	-	-	-	-	-	2	-

LIST OF SAMPLE PROGRAMS

<table>
<tr><th>S.No</th><th>PROGRAMS</th><th>Course Outcomes</th></tr>
<tr><td>1</td><td>Implement basic shell commands.</td><td>CO1</td></tr>
<tr><td>2</td><td>Shell programming: Simple logic programs.
i) Write a menu driven script using the select statement to print calories for food items such as pizza, burger, Salad, Pasta etc.
ii) Write a shell script that, given a filename as the argument will count vowels, blank spaces, characters, number of line and symbols.</td><td>CO1</td></tr>
<tr><td>4</td><td>CPU Scheduling Algorithms
i) A washing machine which requires the process to be executed sequentially. Consider the processes P1, P2, P3, P4 whose arrival times are 1, 5, 9, 10 and burst times are 4, 3, 5, 2 respectively. Implement an appropriate algorithm. Find the CPU idle time, so that the water can be supplied during that period of time.
ii) Implement shortest job first for the following data:
Consider the following set of processes, CPU burst time, Arrival time. Calculate the average waiting time, average response time and average turnaround time.
<table>
<tr><th>Process</th><th>Burst Time</th><th>Arrival Time</th></tr>
<tr><td>P1</td><td>3</td><td>0</td></tr>
<tr><td>P2</td><td>6</td><td>2</td></tr>
<tr><td>P3</td><td>4</td><td>4</td></tr>
<tr><td>P4</td><td>5</td><td>6</td></tr>
<tr><td>P5</td><td>2</td><td>8</td></tr>
</table>
iii)Implement Round Robin for the following data
Consider the following set of processes and length of the CPU burst time given in milliseconds.
<table>
<tr><th>Process</th><th>Burst Time</th></tr>
<tr><td>P1</td><td>10</td></tr>
<tr><td>P2</td><td>1</td></tr>
<tr><td>P3</td><td>2</td></tr>
<tr><td>P4</td><td>1</td></tr>
<tr><td>P5</td><td>5</td></tr>
</table></td><td>CO3</td></tr>
<tr><td>5</td><td>Develop a program to provide synchronization among the producer and consumer processes in producer –consumer problem using a monitor.</td><td>CO3</td></tr>
<tr><td>6</td><td>Consider the following data:
<table>
<tr><th>Process</th><th>Allocation
A B C D</th><th>Max
A B C D</th><th>A B C D</th></tr>
<tr><td>P1</td><td>0 0 1 2</td><td>0 0 1 2</td><td rowspan="5">2 1 0 0</td></tr>
<tr><td>P2</td><td>2 0 0 0</td><td>2 7 5 0</td></tr>
<tr><td>P3</td><td>0 0 3 4</td><td>6 6 5 6</td></tr>
<tr><td>P4</td><td>2 3 5 4</td><td>4 3 5 6</td></tr>
<tr><td>P5</td><td>0 3 3 2</td><td>0 6 5 2</td></tr>
</table>
Calculate the need matrix
Is this system currently in a safe or unsafe state?
Is the system currently deadlock or not.
Which process, if any, or may become deadlocked?</td><td>CO3</td></tr>
<tr><td>7</td><td>Consider the following scenario: A process has been allocated 3 page frames. Assume that none of the pages of the process are available in the memory initially. The process makes the following sequence of page references (reference string): 1, 2, 1, 3, 7, 4, 5, 6, 3, 1, 2, 4, 6, 3,1.
Find a page replacement policy which gives the least number of page faults.</td><td>CO4</td></tr>
<tr><td>8</td><td>Implement the first fit and best fit algorithm in memory management.</td><td>CO4</td></tr>
</table>

9	Implement the Contiguous file allocation method.	**CO4**
10	Implement a bit map for the following scenario: For a memory of size 32 blocks ,the allocated blocks are 2,3,4,5,8,9,10,11,12 and display the bitmap pattern.	**CO4**

REFERENCES:

1. Sumitabha Das, "Unix Concepts and Applications", 4th Edition. TMH, 2006.
2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", 9th Edition, John Wiley & Sons, 2015.
3. William Stalling, "Operating Systems: Internals and Design Principles", 9th edition, PHI, 2018.

WEB REFERENCES:

1. https://nptel.ac.in/content/storage2/courses/106108101/pdf/PPTs/Mod_13.pdf
2. https://nptel.ac.in/courses/117106113/

OBJECT ORIENTED PROGRAMMING USING JAVA LAB	
23CS4214	**Credits : 1.5**
Instruction : 3 Periods / week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites: Basic programming language

Course objective:

- To design the OOPS concepts and apply them in solving problems
- To learn the principles of Inheritance, Polymorphism and relate to the design of Abstract class.
- To get the knowledge of implementation of packages and Interfaces.
- To acquire knowledge about the usage of Exception Handling in Java Applications.

Course Outcomes:

CO1: Apply basics of JAVA programming and analyze OOPS concepts.
CO2: Modularize the application using packages.
CO3: Apply multithreading and Exception Handling.
CO4: Design application using files.
CO5: Design GUI applications using java AWT and Swing.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	3	2	2	-	2	-	-	-	-	-	-	2	2	-
	2	3	2	2	-	2	-	-	-	-	-	-	2	2	-
	3	3	2	2	-	2	-	-	-	-	-	-	2	2	-
	4	3	3	3	-	3	-	-	-	-	-	-	3	3	-
	5	3	3	3	-	3	-	-	-	-	-	-	3	3	-

LIST OF SAMPLE PROGRAMS

S.No	List of Experiments	CO's
1.	a) Write a java program on displaying data using command line arguments b) Write a java program on demonstrating various operators c) Write a java program for displaying prime numbers	**CO1**
2.	a) Write a java program for finding the maximum element in an array b) Write a java program to input a string from the user and count the number of vowels c) Write a java program to input a string from the user and then reverse it. Display the original and reversed string.	**CO1**
3	a) Define Employee: with attributes id, name and salary. Write a Java program by creating multiple constructors to handle different attributes and display employee details using different constructors b) Implement a Calculator with overloaded methods for Arithmetic operations to handle different data types.	**CO2**
4	a) Write a Java program for Create a Shape base class with calculateArea() method. Derive Rectangle and Circle classes from Shape and Override calculateArea() method in derived classes. Use super keyword to call base class method. b) Define a Shape interface with abstract methods calculateArea() and calculatePerimeter(). Create rectangle and circle classes by implementing a Shape interface to calculate area and perimeter for each shape and display results.	**CO2**
5	Create a method that attempts to divide two numbers, and handle ArithmeticException and ArrayIndexOutofBoundsException using try-catch blocks.	**CO3**
6	a) Java Program for Custom Exception Handling. Define a custom exception class named InvalidAgeException that extends Exception. b) Write a program to input user's age and validate it using exception handling. c) Write a Java program for Handling multiple Exceptions	**CO3**
7	Write a Java program to Create two threads: one for printing even numbers and the other for printing odd numbers. Ensure proper synchronization to avoid interleaving output and guarantee correct sequence of numbers.	**CO3**
8	Write a Java program to solve the producer-consumer problem using multithreading and synchronization techniques. Implement a shared buffer to store items produced by a producer thread and consumed by a consumer thread.	**CO3**
9	Write a Java program to read data from an input file using FileInputStream and FileReader, and write data to an output file using FileOutputStream and FileWriter. Demonstrate the usage of SequenceInputStream to concatenate multiple input streams into a single input stream.	**CO4**
10	Design GUI Application Using AWT Components.	**CO5**
11	Design GUI Applications Using Swing Components.	**CO5**
12	Write a java program to handle ActionListener and MouseListener, MouseMotionListener and KeyListener	**CO5**

REFERENCES:

1. P.J. Deitel and H.M. Deitel, "Java for Programmers", Pearson education (OR) P.J. Deitel and H.M. Deitel, "Java: How to Program", PHI.
2. P. Radha Krishna, "Object Oriented Programming through Java", Universities Press.
3. Bruce Eckel, "Thinking in Java", Pearson Education
4. Bruce Eckel, "Programming in Java", Pearson Education

WEB REFERENCES:

1. **https://www.decodejava.com**
2. **http://java.candidjava.com/tutorial/Core-Java-tutorial-with-example.html**
3. **https://docs.oracle.com/javase/tutorial/java/nutsandbolts**
   https://developer.ibm.com/tutorials/j-introtojava1

UI/UX DESIGN TOOLS

23CS9211	**Credits : 1**
Instruction : 2 Periods / Week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites:

- Basic Knowledge of Web design.

Course Objectives:

The course should enable the students to:

- Analyse the significance of UI/UX design.
- Conduct user research using interviews, surveys, and observations
- Implement wireframing and prototyping techniques.
- Comprehend typography, colour theory, implementation of interactive UI elements.

Course Outcomes:

By the end of the course, the student will be able to

CO1: Ability to generate innovative ideas to design solutions.

CO2: Apply the visual design tools to create user interfaces.

CO3: Develop proficiency in designing user experiences for mobile and websites.

CO4: Evaluate the UI/UX designs.

CO	**PO**												**PSO**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	2	3		3	-	-	-	-	3	2	2	2	2	-
CO2	2	-	3	2	3	-	-	-	3	2	2	2	2	-
CO3	2	-	3	2	3	-	-	-	3	2	2	2	2	-
CO4	2	-	-	2	-	-	-	-	3		2	2	2	-

S.No	**Lab Experiments**	**CO's**
1	**Ideation & Design:** • Conduct a brainstorming session to generate design ideas for any problem statement. • Encourage students to explore divergent thinking and come up with innovative solutions. **UX and Visual Design:** • Analyse and critique existing user interfaces from a UX and visual design perspective. • Identify usability issues, visual hierarchy, and alignment with design principles.	**CO1**

2	**Visual Design Tools:** • Introduce students to a visual design tool such as Adobe Photoshop or Illustrator. • Provide hands-on exercises to familiarize them with basic tools and techniques for creating visual designs. **Colour Theory:** • Explore the principles of colour theory and its application in UI/UX design. • Assign exercises to create colour palettes, apply colour psychology to design decisions, and experiment with colour combinations. **Typography:** • Dive into the world of typography and its role in UI/UX design. • Assign exercises to practice pairing fonts, adjusting typography for readability, and creating typographic hierarchy. **Prototyping & Interactions:** • Introduce students to prototyping tools such as Adobe XD, Figma. • Guide them through creating interactive prototypes with clickable elements, transitions, and animations. **Portfolio Design:** • Guide students in designing their professional portfolio to showcase their UI/UX design projects. • Provide templates for organizing and presenting their work effectively.	**CO2**
3	**UX Design for Mobile App:** • Assign a project to design the user experience for a mobile app. • Guide students through user research, wireframing, prototyping specific to mobile platforms. **UX Design for Website:** • Assign a project to design the user experience for a website. • Cover topics such as information architecture, navigation design, responsive layout, and user testing for web interfaces. **Graphic Designing:** • Explore advanced graphic design techniques relevant to UI/UX design. • Create custom icons, and visual elements for digital interfaces.	**CO3**
4	**Usability Testing:** • Conduct usability testing sessions with users to evaluate the effectiveness of a UI/UX design. • Guide students through creating test scenarios, moderating sessions, and analysing feedback. **Wireframing:** • Assign a project to create wireframes for a digital interface. • Emphasize the importance of information hierarchy, content layout, and user flow in wireframe design. **Case Studies:** • Analyse real-world case studies of successful UI/UX design projects. • Assign students to research and present case studies, highlighting key design principles, challenges, and outcomes.	**CO4**

Text Books:

1. Don Norman, "The Design of Everyday Things", Basic Books, Revised and Expanded Edition.
2. Steve Krug, "Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability", New Riders, Third Edition.
3. Jesse James Garrett, "The Elements of User Experience: User-Centered Design for the Web and Beyond", New Riders, Second Edition.

Reference Books:

1. Steve Portigal, "Interviewing Users: How to Uncover Compelling Insights", Rosenfeld Media, Second Edition.
2. Susan Weinschenk, "100 Things Every Designer Needs to Know About People", New Riders, Second Edition.

WEB REFERENCES:

1. "Designing Interfaces: Patterns for Effective Interaction Design" by Jenifer Tidwell. Available athttps://www.designinginterfaces.com/
2. "Lean UX: Designing Great Products with Agile Teams" by Jeff Gothelf and Josh Seiden. Available at https://www.oreilly.com/library/view/lean-ux/9781491953600/

Logical Reasoning & Corporate Skills

(II Year, I Sem.)

Course Category	Humanities	**Credits**	1
Branch	All Branches		
Course Code	23TP9101	**Lecture-Tutorial - Practical**	2+2
Pre requisites	Knowledge of LSRW Skills, Basic Maths	**Continuous Evaluation**	
		Semester End Evaluation	
		Total Marks	100

Upon successful completion of the course, the student will be able to:

Course Outcomes	**CO1**	Enforce corporate etiquette, and precise usage of English grammar to enhance their professional communication. L3
	CO2	Master negotiation skills and telephone etiquette with emotional intelligence for corporate interactions.L3
	CO3	Enhance email writing skills by incorporating vocabulary acquired from storytelling, situational dialogues and reading activities by using various digital tools. L3
	CO4	Use their logical thinking and analytical abilities to solve reasoning questions from number analogy and series and letter based aptitude questions company specific and other competitive tests.
	CO5	Solve questions related to clock and calendar, etc.. from company specific and other competitive tests.

Contribution of Course Outcomes towards achievement of Program Outcomes& Strength of correlations		**PO1**	**PO2**	**PO3**	**P04**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
	CO1									M	M		M		
	CO2									M	M		M		
	CO3									M	M		M		
	CO4	M													
	CO5	M													
	L- Low					**M-Medium**								**H-High**	

PART-A: Verbal Ability Skills

UNIT-1	Corporate Etiquette – Work Place Etiquette and Conflict Resolution - Grammar Revision Verbal Ability : Prepositions, Articles, tenses and conjunction	**CO1**
UNIT-2	EQ – Negotiation Skills – Telephone Etiquette – MNCs Paper Model Introduction, Situational Dialogue Practice – Team Activities Related to Spoken English Verbal Ability: Fill in the blanks (Based on the given appropriate words)	**CO2**
UNIT-3	E Mail Writing – Vocabulary from Story Telling Activity –MNCs Model Paper 1 Practice Verbal Ability: Sentence arrangements	**CO3**
UNIT-4	Virtual Reading – Functional English – IELTS Vocabulary – News Paper Reading Verbal Ability: Inferred meaning (Homophone, Homonyms)	**CO3**

PART-B: Logical Reasoning	
UNIT-I: Numericalcomputation: Number Series, Letter Series, Number analogy, letter analogy, word analogy	**CO4**
UNIT-II: Coding Decoding- Letter to letter, letter to digit, letter to number and symbol, Word to word coding, odd man out	**CO4**
UNIT-III: Directions-Finding distance, Direction and Shadow based problem, Blood Relations-Mixed Blood Relations, Puzzle-Based Blood Relation, Single-Person Blood Relation, Symbol based Blood Relations.	**CO4**
UNIT-IV: Clocks –finding Angle, Time, Mirror image, Faulty clock, Calendars – Finding day of the week, Number of odd days, Repetition of same calendar	**CO5**
UNIT-V: Seating Arrangement-Circular arrangement, linear arrangement, Order Sequence and Ranking	**CO5**

FINANCIAL LITERACY							
Code	Periods			Sessional Marks	End Exam Marks	Total Marks	Credits
	L	T	P				
23MC0103	30	-	-	100	-	100	-

Prerequisite:

Course Objectives: The course has been designed to give familiarity with different aspects of financial literacy such as savings, investment, taxation, and insurance and understand the relevance and process of financial planning.

Course Outcomes: At the end of the course the student will be able to:	
CO-1	Recognize the role of saving money in reaching financial goals and identify components of a spending plan .
CO-2	Describe the importance of banks and their purpose as financial institutions.
CO-3	Apply the concept of investment planning.
CO-4	Ability to analyse banking and insurance products.
CO-5	Estimate Personal tax.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO-1						1		1			2	2
CO-2						1		1			2	2
CO-3						1		1			2	2
CO-4						1		1			2	2
CO-5						1		1			2	2

Course Outcomes	PSO1	PSO2
CO-1		
CO-2		
CO-3		
CO-4		
CO-5		

CO- Course Outcome; PO- Program Outcome; PSO-Program Specific Outcome; Level- 1: Low, 2: Medium, 3: High

SYLLABUS	
UNIT - I	**Periods: 4L+2T=6**
UNIT TITLE:Introduction to Financial Planning	
Introduction to saving: Benefits of Savings-Saving vs Investment, Investment vs Gambling-Time value of money-Management of spending and financial discipline.	
UNIT - II	**Periods: 4L+2T=6**
UNIT TITLE:Banking and Digital Payment	
Banking products and services -Savings account, Current Account, Fixed deposits, Recurring deposits-Digitisation of financial transaction- Modes of digital payments: Debit cards, Credit cards, Net banking and UPI,-Digital Wallets-Role of RBI in banking sector.	
UNIT - III	**Periods: 4L+2T=6**
UNIT TITLE:Financial Markets and Investment Planning	
Financial Markets: Primary and Secondary markets- Securities and its types, i.e., Equity, Debentures or Bonds, IPOs and FPOs-Mutual Funds: Types of Mutual Funds-Stock Market, DEMAT.	
UNIT - IV	**Periods: 4L+2T=6**
UNIT TITLE:Insurance Services: Life Insurance	
Policies- Term insurance, Endowment policies, Pension policies-Health Insurance Plans-ULIP-General Insurance-Understanding of Ponzi Schemes.	
UNIT - V	**Periods: 4L+2T=6**
UNIT TITLE:Personal Tax	
Introduction to basic tax structure in India for personal taxation-Basic concepts of Income Tax- Exemption and Deduction for individual-Income Tax Act, 1961-E-Filling.	
TEXT BOOKS:	
1.	Introduction to Financial Planning (4th Edition 2017)- Indian Institute of Banking & Finance.
2.	Sinha, Madhu. Financial Planning: A Ready Reckoner July 2017, McGraw Hill.
REFERENCE BOOKS:	
1.	Halan, Monika, Lets Talk Money: You've Worked Hard for It, Now Make It Work for You, July 2018 Harper Business.
2.	Pandit, Amar The Only Financial Planning Book that You Will Ever Need, Network 18 Publications Ltd.
WEB RESOURCES:	
1.	https://onlinecourses.nptel.ac.in/noc21_mg40/preview
2.	https://corporatefinanceinstitute.com/resources/management/financial-literacy/

Semester –II

PROBABILITY & STATISTICS
(Common to CSE, CSD, CSM & IT)

23MA1107 **Credits:3**

Instruction : 3 periods & 1 Tutorial/Week Sessional Marks:40
End Exam : 3 Hours End Exam Marks:60

Prerequisites: Elementary knowledge of set theory, combinations and basic statistics.

Course Objectives:

To provide the required mathematical support in real life problems and develop probabilistic models which can be used in several areas of science and engineering.

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

1.	Analyze the basic principles of statistical measures and probability.
2.	Demonstrate a random variable that describe randomness or uncertainty in certain realistic situation.
3.	Differentiate the concepts of discrete, continuous probability distributions and able to solve problems of probability.
4.	Evaluate simple correlation between the two variables and fit curves by the method of least square approximation.
5.	Analyze the statistical data and apply various small and large samples tests for testing the hypothesis.

CO-PO –PSO Mapping:

CO	PO												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2										1			
CO2	3	2										1			
CO3	3	2										1			
CO4	3	2										1			
CO5	3	2										1			

Correlation levels1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Mapping of Course Outcomes with Program Outcomes &Program Specific Outcomes:

	CO-PO-PSO Justification
1	CO 1 deals with the knowledge of statistical measures and probability is widely used for algorithm development, data analysis, machine analysis and simulation modelling.
2	CO 2 deals with the knowledge of random variable is a vital tool in machine learning. It creates functions for understanding possible outcomes.
3	CO 3 deals with the knowledge of probability distributions is widely used in speech recognition, robotics and network traffic etc.
4	CO 4 deals with the knowledge of curve fitting is widely used as an aid for data visualization and regression is to summarize the relationship among two or more variables.
5	CO 5 deals with the knowledge of testing of hypothesis is mainly used for making statistical decision using experimental data in various filed of computer science.

SYLLABUS

UNIT I **10 Periods**

STATISITCAL METHODS AND DATA ANALYSIS

Measures of central tendency : Mean – Median – Mode.

Measures of dispersion : Mean deviation – Standard deviation – Variance.

PROBABILITY

Introduction to Probability : Definition of random experiment – Events and sample space – Definition of probability – Addition and multiplication theorems – Conditional probability – Baye's theorem – Simple problems on Baye's theorem.

UNIT II **10 Periods**

RANDOM VARIABLES

Discrete and continuous random variables – Distribution function of random variable – Properties, Probability mass function, Probability density function – Mathematical expectation – Properties of mathematical expectation – Moments – Moment generating function – Mean and variance – Simulation of random variable – Solving problems by using Monte Carlo method.

UNIT III **10 Periods**

PROBABILITY DISTRIBUTIONS

Discrete Distributions: Binomial distribution – Poisson distribution – Mean, Variance,

Moment Generating function and problems.

Continuous Probability Distributions: Uniform distribution – Exponential distribution, Memoryless property – Normal distribution – Properties of normal distribution – Importance of normal distribution – Area properties of normal curve – MGF – Mean ,variance and simple problems.

UNIT IV **10 Periods**

CORRELATION, REGRESSION ANALYSIS AND CURVE FITTING

Correlation : Definition – Karl pearson's coefficient of correlation – Measures of correlation – Rank correlation coefficients.

Regression : Simple linear regression – Regression lines and properties.

Curve Fitting : Principle of least squares – Method of least squares – Fitting of straight lines – Fitting of second degree curves and exponential curves.

UNIT V **10 Periods**

TESTING OF HYPOTHESIS

Introduction – Null hypothesis – Alternative hypothesis – Type – I , II errors – Level of significance – Critical region – Confidence interval – One sided test – Two sided test.

Small Sample Tests : Students t - distribution and its properties – Test of significance difference between sample mean and population mean – Difference between means of two small samples – F- Distribution – Test of equality of two population variances – Chi-square test of goodness of fit .

Large sample Tests : Test of significance of large samples – Tests of significance difference between sample proportion and population proportion & difference between two sample proportions – Tests of significance difference between sample mean and population mean & difference between two sample means.

TEXT BOOK :

T. Veerarajan, *Probability, Statistics and Random Processes,* Tata McGraw Hill Publications.

REFERENCE BOOKS:

1. **Kishor S. Trivedi**, *Probability & Statistics with Reliability, Queuing and Computer Applications*, Prentice Hall of India .
2. **B. S. Grewal**, *Higher Engineering Mathematics*, 43rd edition, Khanna publishers, 2017.
3. **Sheldon M.Ross,** *Probability and Statistics for Engineers and Scientists*, Academic Press.
4. **S C Gupta and V.K.Kapoor**, *Fundamentals of Mathematical Statistics*.

FORMAL LANGUAGE AND AUTOMATA THEORY	
23CS4115	**Credits: 3**
Instruction: 3 Periods & 1 Tut/Week	Sessional Marks: 40
End Exam: 3 Hours	End Exam Marks: 60

Prerequisites:

- The students are expected to have a strong background in the fundamentals of discrete mathematics.
- Some knowledge of programming languages, programming, and computer architecture will be helpful.

Course Objectives:

1. Learning the fundamentals of Regular and Context Free Grammars and Languages
2. Introducing the relation between Regular Language and Finite Automata and machines
3. Designing Automata's and machines as Acceptors and Verifiers
4. Distinguish the relation between Contexts free Languages, PDA and TM
5. Introduction about Post Correspondence Problem

Course Outcomes:

At the end of the course students able to

1. Design deterministic and non-deterministic machines
2. Development of regular languages and regular expressions
3. Describe and construct Context Free Grammar
4. Develop pushdown automata accepting strings and Turing machine
5. Summarize about Chomsky hierarchy and distinguish between decidability and undecidability problems

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	3	3	2	3	-	-	-	-	-	-	-	2	2	-
	2	2	2	3	2	-	-	-	-	-	-	-	2	2	-
	3	2	2	3	2	-	-	-	-	-	-	-	2	2	-
	4	2	3	3	2	-	-	-	-	-	-	-	2	2	-
	5	2	3	3	3	-	-	-	-	-	-	-	2	2	-

SYLLABUS

UNIT-I: **12 periods**

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) - Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. **NONDETERMINISTIC FINITE AUTOMATA (NFA)**- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

Learning Outcomes:

1. An ability to design grammars and automata (recognizers) for different language classes.

2. An ability to prove and disprove theorems establishing key properties of formal languages and automata**.**

UNIT-II: **12 periods**

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA to Regular Expressions, Converting Regular Expressions to Automata, and applications of Regular Expressions.
REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, and Closure properties of regular languages.

Learning Outcomes:

1. Design Finite Automata for different Regular Expressions and Languages.

2. Illustrate the Pumping lemma for proving that languages are not regular.

UNIT-III: **12 periods**

CONTEXT FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted).

Learning Outcomes:

1. To construct context free grammar for various languages.

2. Describe the language accepted by automata or generated by a regular expression or a context-free grammar.

UNIT-IV: **12 periods**

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. **TURING MACHINES (TM)**: Formal definition and behavior, Languages of a TM, TM as accepters and as a computer of integer functions, Types of TMs.

Learning Outcomes:

1. To solve various problems of applying normal form techniques, push down automata and Turing Machines.

2. Construct a pushdown automaton for a given context-free language

UNIT-V: **12 periods**

RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL):
Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

Learning Outcomes:

1. Analyzingthe basic results on computability, including undecidable problems such as the halting and Post Correspondence problems, and their significance.
2. Define the various categories of languages and grammars in the Chomsky hierarchy.

TEXT BOOKS

1. Hopcroft H.E. and Ullman J. D. "Introduction to Automata Theory Languages and Computation", Pearson Education

REFERENCES

1. Theory of Computer Science – Automata languages and computation -Mishra and Chandrasekharan, 2nd edition, PHI.
2. Introduction to languages and the Theory of Computation, John C Martin, TMH, 4th edition.

WEB REFERENCES

1. https://nptel.ac.in/courses/111103016
2. https://nesoacademy.org/cs/04-theory-of-computation
3. https://notesioe.com/toc-notes-2/

DATABASE MANAGEMENT SYSTEMS	
23CS4116	**Credits: 3**
Instruction: 3 Periods & 1 Tut/Week	Sessional Marks: 40
End Exam: 3 Hours	End Exam Marks: 60

Pre-requisites:

- Basic Knowledge about Set Theory, Relations and Functions and Data Structures.

Course Objectives:

- To acquire knowledge on fundamental concepts such as data models, schemas, relational algebra, normalization.
- Develop skills in writing and executing queries using Structured Query Language
- for data retrieval, manipulation, and management.
- Analyze a given database application scenario to use ER model for conceptual design
- of the database.
- Apply normalization techniques to improve database design.
- To have a good understanding of how transactions are managed in databases, including concurrency control mechanisms to maintain data consistency.

Course Outcomes (CO):

By the end of the course, the student will be able to:	
CO-1	Describe basic concepts of database systems and principles of transaction processing, concurrency techniques and recovery of database.
CO-2	Apply Conceptual and logical database design principles, including E-R diagrams.
CO-3	Compose SQL queries to perform operations on database (Create, Retrieve, Update, Delete), Relational Algebra and Relational Calculus queries.
CO-4	Analyze and apply schema Refinement, database normalization principles.
CO-5	Demonstrate the concept of Transaction Processing and Concurrency control.

Mapping of Course Outcomes with Program Outcomes:

CO	PO												PSO	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
1	3	3	3	3	3	3	-	-	3	3	-	3	3	-
2	3	3	3	2	2	3	-	-	3	2	-	2	2	-
3	3	2	3	3	3	3	-	-	2	3	-	3	3	-
4	3	3	2	2	3	2	-	-	2	3	-	3	3	-
5	3	2	3	3	3	3	-	-	3	2	-	2	2	-

SYLLABUS

UNIT-I: **12 Periods**

Introduction to DBMS: Overview of Database Systems, File system versus a DBMS, Advantages of a DBMS, Database applications , Three tier Schema architecture of DBMS, Schemas and Instances,Data Models, Database Languages, Transaction Management, Structure of a DBMS, Client/Server Architecture, Database Administrator and Users.

Learning Outcomes: At the end of this unit the students will be able to:

1. To gain knowledge about the working of a simple Database System.
2. Describe the fundamental concepts and principles of Database Management Systems, enabling them to effectively design, implement, and manage databases in various real-world scenarios

UNIT-II: **12 Periods**

Entity-Relationship Model: Design Issues, Representation of entities, attributes, entity set, relationship, relationship sets, Cardinality constraints, Weak-entity types, Subclasses and inheritance, Specialization, Generalization, Aggregation, Conceptual Database Design withthe ER Model.

Relational Model: Structure of Relational Databases, Basics of Relational Model, IntegrityConstraints, Logical Database Design, Introduction to Views, Destroying/ Altering Tables and Views.

Learning Outcomes: At the end of this unit the students will be able to

1. Describe the fundamental elements of relational database management systems and design ER-models to represent simple database application scenarios.
2. Design entity relationships and convert entity relationship diagrams into RDBMS.

UNIT-III: **12 Periods**

Relational Algebra: Selection and Projection, Set Operations, Renaming, Joins, Division

Relational Calculus: Tuple Relational Calculus, Domain Relational Calculus

Queries, Constraints, Triggers: Concepts of DDL, DQL, DML, DCL, TCL Commands, The Form of Basic SQL Query, Union, Intersect, and Except, Nested Queries,Correlated Nested Queries, Set- Comparison Operators**,** Aggregate Operators, Null Values, Complex Integrity Constraints in SQL, Triggers and Active Databases.

Learning Outcomes: At the end of this unit the students will be able to

1. Develop a conceptual understanding of relational calculus as a formal language for expressing queries.
2. Implement the basics of SQL and construct queries using SQL in database creation and interaction.

UNIT-IV: **12 Periods**

Database Design: Schema Refinement, Functional Dependencies, Reasoning about Functional Dependencies, Normalization using functional dependencies (1NF, 2NF and 3NF), Boyce-Codd normal form(BCNF), Lossless join and dependency preserving decomposition, Multi-valued dependencies and Fourth normal form(4NF).

Learning Outcomes: At the end of this unit the students will be able to

1. Understands the importance of schema refinement and applies Normalization techniques to improve database design.

UNIT-V: **12 Periods**

Transaction Management: **Transaction Management**: Transaction States, ACID Properties, Transactions & Schedules, and transaction management with SQL using commit rollback and savepoint, Concurrent Execution of Transactions, Lock- Based Concurrency Control.

Concurrency Control: 2PL, Serializability and Recoverability, Concurrency control with time stamp ordering: Wait/Die and Wound/Wait Schemes.

Learning Outcomes: At the end of this unit the students will be able to:

1. Execute various SQL queries related to Transaction Processing & Locking using the concept of Concurrency control.
2. Apply knowledge of timestamp ordering and conflict resolution schemes to design and implement efficient concurrency control mechanisms in a DBMS.

Textbooks:

1. Database Management Systems, by Raghu Ramakrishnan, Johannes Gehrke, McGraw- Hill Third edition, 2014.
2. Database System Concepts, by A.Silberschatz.H.Korth , McGraw-Hill,7th edition 2019.

References Books:

1. Introduction to Database Systems, by Bipin Desai, Galgotia Publications, 2015.
2. Fundamentals of Database System, by RamezElmasri, Shamkant B. Navathe, Pearson Education 7th edition 2015.

Web Resources:

1. https://www.oreilly.com/library/view/web-database-
2. https://www.edx.org/learn/databases

COMPUTER ORGANIZATION AND MICROPROCESSOR INTERFACING	
23CS4117	**Credits: 3**
Instruction: 3 Periods & 1 Tut/Week	Sessional Marks: 40
End Exam: 3 Hours	End Exam Marks: 60

Prerequisites: Basic knowledge of Digital Logic Design

Course Objectives:

- To study the basic structure and operation of digital computer and design of arithmetic and logic unit.
- To acquire knowledge on design the microprogrammed control unit and external peripheral device.
- Familiar with the architecture and the instruction set of an Intel microprocessor 8085 and 8086.
- Learning the concept of interfacing various I/O peripherals like Keyboard/Display with microprocessors using 8255 PPI
- To provide solid foundation on interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real time problems

Course Outcomes:

By the end of the course, the student will be able to:	
1.	Analyze the Computer Instructions & Memory Instructions and working of ALU circuit and its operations.
2.	Design the working principles of hardwired and microprogrammed control unit and external peripheral devices.
3.	Demonstrate the architecture and pin configuration of 8085 Microprocessors and the significance of Addressing modes, timing diagrams and analyze the working of 8085 interfacing with co-processors are 8255 &8279,
4.	Demonstrate the programming knowledge for practical implementation of assemble level programming using instruction set of 8085.
5.	Describe the architecture and pin configuration of 8086 Microprocessors and practical implementation of assemble level programming using instruction set of 8086.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	2	2	3	-	-	-	-	-	-	-	-	-	2	-
	2	3	2	2	-	-	-	-	-	-	-	-	-	2	-
	3	2	2	2	-	2	-	-	-	-	-	-	2	2	-
	4	2	2	2	-	-	-	-	-	-	-	-	-	2	-
	5	2	2	3	-	2	-	-	-	-	-	-	2	2	-

SYLLABUS

UNIT- I

Basic computer Organization **12 periods**

Register Transfer Language, Bus and Memory Transfers, Arithmetic Logic Shift Unit. Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions.

Learning Outcome: At the end of this Unit the students will be able to

1. Identify the basic principles of a computer, Computer Instructions & Memory Instructions and working of ALU circuit and its operations.
2. Analyze the computer Instruction, Instruction codes, Instruction Cycle and examine the procedure of an Instruction Cycle.

UNIT -II **12 periods**

Control Design: Hardwired & Micro Programmed (Control Unit), Control Memory, Address Sequencing, Micro program Example.

Input-Output Organization

Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.

Learning Outcome: At the end of this Unit the students will be able to

1. Specify the design of a control unit in a computer.
2. Distinguish between Hardwired & Micro programmed control unit.
3. Deduce the design issues of Input-output organization

UNIT -III **12 Periods**

Introduction to Microprocessors and Microcomputers, Internal Architecture and Functional Description of typical 8-bit 8085 μP, Instruction Set, types of Instructions, addressing modes of 8085 and Timing Diagrams of 8085 μP. **Interfacing Peripheral ICs to Intel 8085:** Programmable peripheral interface (8255A), Programmable Keyboard / Display controller (8279).

Learning Outcome: At the end of this Unit the students will be able to

1. Describe the basic architecture of 8085 and the functional description of 8085, and list the Instruction set; state the addressing modes and timing diagrams of 8085.
2. Illustrate the different peripherals (8255, 8279.) are interfaced with Microprocessors, and also describe the control word formats of all Programmable peripheral interfaces

UNIT -IV **12 Periods**

8085 Assembly Language Programming

Introduction to Assembly Language Programming Techniques: Looping, Counting, and Indexing, Counter and timing Delays, Stack and Subroutines, Code Conversions, BCD Arithmetic operations, 16-bit data Operations, Interrupts and Interrupt Service Routines.

Learning Outcome: At the end of this Unit the students will be able to

1. Develop the assembly language programs using various programming techniques.
2. Analyze the simple programs of call instructions, sorting, and string manipulations.

UNIT-V **12 Periods**

Introduction to 8086 Microprocessor

Internal Architecture and Functional/Signal Description of 8086/8088, Segmented Memory, Maximum-Mode and Minimum-Mode Operation and Addressing Modes of 8086, **Assembly Language Programming:** 8086 Instruction Set, Interrupts and Interrupt Service Routines, Assembler Directives.

Learning Outcome: At the end of this Unit the students will be able to

1. Describe the modes and functional block diagram of 8086 along with pins and their functions.
2. Used the programming techniques of 8086 to build programs using instruction set.
3. List, describe and use different types of instruction, directives and interrupts.

TEXT BOOKS:

1. M.Morris Mano, ―Computer System Architecture‖, Pearson Education Inc., 2003, Third Edition.
2. Ramesh S. Gaonkar, "Microprocessor Architecture, Programming, and Applications with the 8085" Penram International, 6th Edition.
3. John E.Uffenbeck, "The 80x86 Family, Design, Programming and Interfacing 3rdEdition, Pearson Education Inc.", 2002.

REFERENCE BOOKS:

1. William Stallings, Computer Organization and Architecture, 6th Edition, Pearson/PHI, 2007.
2. Andrew S. Tanenbaum, Structured Computer Organization, 5th Edition, PHI/Pearson, 2007.
3. BARRY B. BREY, "The Intel Microprocessors 8086/8088, 80186/80188,80286,80386 and 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, Architecture, Programming and Interfacing", Pearson Education Inc., 2003,6thEdition.
4. Douglass V. Hall, "Microprocessors and Interfacing, Programming and Hardware", TMH Edition, 1999, 2ndEdition

ONLINE WEB RESOURCES:

1. https://nptel.ac.in/courses/106/103/106103068/
2. https://swayam.gov.in/nd1_noc20_ee11
3. https://www.udemy.com/course/microprocessor_8085/
4. https://www.udemy.com/course/interfacing-8086-microprocessor-with-peripheral-devices/

DESIGN AND ANALYSIS OF ALGORITHMS	
23CS4118	**Credits : 3**
Instruction **:**3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites: Basic Programming Language, Problem Solving and Data Structures.

Course Objectives:

1. Analyze the asymptotic performance of algorithms.
2. Demonstrate a familiarity with major algorithms and data structures.
3. Apply important algorithmic design paradigms and methods of analysis.
4. Synthesize efficient algorithms in common engineering design situations.

Course Outcomes:

1. Analyze running times of algorithms using asymptotic analysis.
2. Outline students with specific algorithms for a number of important computational problems like sorting, searching and graphs, etc.
3. Apply important algorithmic design paradigms and method of analysis.
4. Combine efficient algorithms in common engineering design situations.
5. Explain the concept of P, NP, and NP-complete problems and prove that a certain problem is NP-complete.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	2	3	2	2	-	-	-	1	1	2	-	-	2	2
	2	3	3	3	2	-	-	-	1	1	2	-	3	2	2
	3	2	3	3	3	-	-	-	1	1	2	-	3	2	2
	4	2	3	2	2	-	-	-	1	1	2	-	3	2	2
	5	2	3	2	3	-	-	-	1	1	2	-	1	2	2

SYLLABUS

UNIT I **12 Periods**

Introduction:

Problem solving steps and Problem Types, Asymptotic Notations and Efficiency Classes, Mathematical Analysis for Recursive Algorithms and Non- Recursive Algorithms.

Learning outcomes: At the end of this unit, students are able to

1. Discuss the correctness of algorithms using inductive proofs and invariants.
2. Analyze worst-case running times of algorithms using asymptotic analysis.

UNIT II **12Periods**

Brute Force: Selection and Bubble sort, Sequential Search, Closest- Pair problem.

Exhaustive Search: Travelling Salesman Problem, Knapsack Problem, Assignment Problem.

Divide-and-Conquer: General Method, Binary Search, Merge Sort, Quick Sort, Stassen's Matrix Multiplication.

Learning outcomes: At the end of this unit, students are able to

1. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it.
2. Apply algorithms that employ this paradigm.

UNIT III **12 Periods**

Transform and Conquer: Pre-sorting, Gauss Elimination, Balanced Trees: 2-3 Trees, Heap Sort, Problem Reduction: Least Common Multiple.

Dynamic Programming: All Pair Shortest Path Problem, Optimal Binary Search Trees, 0/1 Knapsack Problem.

Learning outcomes: At the end of this unit, students are able to

1. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it.
2. Apply algorithms that employ this paradigm. Synthesize dynamic programming algorithms and analyze them.

UNIT IV **12 Periods**

Greedy Technique: Minimum Spanning Tree Algorithm, Single Source Shortest path Problem, Huffman Trees.

Space and Time Trade-offs: Sorting by computing, Input Enhancement in String Matching- Horspool's Algorithm, Boyer-Moore Algorithm

Backtracking: N-Queen's problem, Sum of subsets problem, Graph Coloring

Learning outcomes: At the end of this unit, students are able to

1. Apply the greedy approach to solve the MST and Shortest path.
2. ImplementBacktracking techniques.

UNIT V **12Periods**

Branch and Bound: General method, Applications: Travelling salesman problem, 0/1 knapsack problem, Assignment Problem.

NP-HARD & NP-COMPLETE PROBLEMS: P, NP, NP - Hard and NP Complete classes.

Learning outcomes: At the end of thisA unit, students are able to

1. Apply Branch and bound approach to solve TSP & Knapsack problems.
2. Perform competitive analysis.

TEXT BOOKS

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2012
2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", Third Edition, PHI Learning Private Limited, 2012.

REFERENCES

1. "The Design and Analysis of computer Algorithms", by Aho, Hopcroft & Ullman, Pearson Education.
2. J. Kleinberg and E. Tardos. Algorithm design. Addison-Wesley. First edition 2005, 2nd edition, February 2022.

WEB RESOURCES

1. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-toalgorithms-spring-2008/lecture-notes/
2. https://www2.cs.duke.edu/courses/fall10/cps130/lectures.html
3. https://www.isical.ac.in/~arijit/courses/spring2017/daa-mtech.html
4. http://www.cs.umd.edu/class/fall2015/cmsc451/
5. http://www.cse.iitd.ernet.in/~ssen/csl356/admin356.html
6. https://www.cs.virginia.edu/~luebke/cs332/
7. https://www.cs.umd.edu/users/samir/

DATABASE MANAGEMENT SYSTEMS LAB	
23CS4216	**Credits: 1.5**
Instruction**:** 3 Periods/week	Sessional Marks: 50
End Exam: 3 Hours	End Exam Marks: 50

Pre-requisites:

- Elementary knowledge about computers including some experience using UNIX or Windows.
- Knowledge about data structures and algorithms, corresponding to the basic course on Data Structures and Algorithms.

Course Objectives:

- To master the basics of SQL and construct queries using SQL.
- To provide an environment that is both convenient and efficient for users to retrieve and store information.

Course Outcomes (CO):

By the end of the course, the student will be able to:	
CO1	Implementation of basic SQL queries.
CO2	Understand various advanced queries execution such as nested queries and joins.
CO3	Construct triggers, views, and stored procedures for different scenarios
CO4	Apply the principles of ER model and normalization for schema refinement in logical database design.

.

Mapping of Course Outcomes with Program Outcomes:

CO	**PO**												**PSO**	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
1	1	2	2	1	2	1	-	-	1	-	-	-	2	2
2	2	3	2	2	2	1	-	-	1	-	-	-	2	2
3	2	3	2	2	2	1	-	-	1	-	-	-	2	2
4	2	3	3	2	2	1	-	-	3	-	-	-	3	2

SYLLABUS

List of Experiments:

Experiment Name	Mapping
1. SQL DDL ,DML Statements	CO1
2. SQL Constraints.	CO1
3. Inbuilt functions in RDBMS.	CO2
4. Aggregate functions	CO2
5. Nested Queries & Join Queries.	CO2
6. Creation and dropping of Views.	CO3
7. Simple PL/ SQL programs , Creating Triggers, Cursors	CO3
8. Normalization.	CO4
9. Developing a sample application which includes all database design steps like requirements analysis, logical database design, normalization, developing user interface to access database from the application.	CO4

Sample Applications:

1. Development of an Online Course Portal for a campus
2. Book Bank Management System
3. Car Rental Management System
4. Exam/academic system for College Management
5. Real estate Management system
6. University Management System
7. Database manager for a Magazine agency or a newspaper agency
8. Ticket booking for performances.
9. Inventory Control System
10. Student management System

Textbooks:

1. Database Management Systems, by Raghu Ramakrishnan, Johannes Gehrke, McGraw- Hill Third edition, 2014.
2. Database System Concepts, by A.Silberschatz.H.Korth , McGraw-Hill,7th edition 2019.

References Books:

1. Introduction to Database Systems, by Bipin Desai, Galgotia Publications, 2015.
2. Fundamentals of Database System, by RamezElmasri, Shamkant B. Navathe, Pearson Education 7th edition 2015.

COMPUTER ORGANIZATION AND MICROPROCESSOR INTERFACING LAB	
23CS4217	**Credits: 1.5**
Instruction**:** 3 Practical /Week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites: Basic knowledge of Digital logic design

Course Objectives:

1. Able to design the simple logic circuits and test/verify the functionality of the logic circuits.
2. Developing of assembly language programs and providing the basics of the processors.
3. To provide solid foundation on interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real time problems
4. To assist the students with sufficient knowledge on the interrupts and working with interrupt driven I/O for communication with external devices.

Course Outcomes:

By the end of the course, the student will be able to:	
1	Identify the basic principles and apply to arithmetic for ALU implementation.
2	Possessed a better command over the instruction of set of 8085 and 8086 microprocessor for programmatically deployment.
3	Demonstrate the interfacing of 8085 microprocessor with external I/O devices through 8255 PPI.
4	Students will possess the knowledge to design and develop a working prototype with various simulators and emulators that they have used throughout the lab sessions.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	2	1	3	-	-	-	-	-	-	1	-	-	-	-
	2	3	1	2	-	-	-	-	-	-	1	-	-	1	-
	3	2	1	2	-	-	-	-	-	-	1	-	-	1	-
	4	1	1	2	-	1	-	-	-	-	1	-	-	-	-

S.NO	LIST OF EXPERIMENTS	No. of Weeks	CO's
1	**COMPUTER ORGANIZATION** Verification of truth table of various logic gates. Study of Arithmetic Logic Unit (ALU)	4	1
2	**8085 ASSEMBLY LANGUAGE PROGRAMMING** According to theory course using the following: Keyboard Monitor of 8085µP Trainer kit.	4	2 &4
3	**INTERFACING WITH 8085 TRAINER** 8255 Study Card Scenarios (I/O and BSR Mode Operations) 8255 Modes Using Hardware Interrupts Traffic Light Controller Stepper Motor Controller Keyboard/Display Interface	4	3
4	**8086 ASSEMBLY LANGUAGE PROGRAMMING** According to theory course using the following: PC Assembler using TASM or MASM, TD or SYMDEB or CVD (Code View debugger)	2	2&4

TEXT BOOKS:

1. M.Morris Mano, "Computer System Architecture", Pearson Education Inc., 2003, Third Edition.
2. Ramesh S. Gaonkar, ―Microprocessor Architecture, Programming, and Applications with the 8085‖ Penram International, 6th Edition.
3. John E.Uffenbeck, ―The 80x86 Family, Design, Programming and Interfacing 3rdEdition, Pearson Education Inc.‖, 2002.

REFERENCE BOOKS:

1. William Stallings, Computer Organization and Architecture, 6th Edition, Pearson/PHI, 2007.
2. Andrew S. Tanenbaum, Structured Computer Organization, 5th Edition, PHI/Pearson, 2007.
3. Walter A. Tribel and Avtar Singh, ‖ The 8088 and 8086 Microprocessors, Programming, interfacing, Software, Hardware, and Applications‖, Pearson Education Inc., 2003, 4thEdition.
4. Douglass V. Hall, ―Microprocessors and Interfacing, Programming and Hardware‖ , TMH Edition, 1999, 2ndEdition

ONLINE WEB RESOURCES:

1. https://swayam.gov.in/nd1_noc20_ee11/preview
2. https://www.coursera.org/lecture/cs-algorithms-theory-machines/digital-circuits-91A4N

Full Stack Web Development

23CS6211	**Credits : 1.5**
Instruction : 3 Periods /Week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites:

- Basic Knowledge of Programming Fundamentals and UI/UX design.

Course Objectives: The course should enable the students to:

- Acquire knowledge on Web page designing.
- Validating the content of Webpage from Client side
- Creating the Single page Applications.
- Develop a Server environment in our machines.

Course Outcomes:

By the end of the course, the student will be able to

1. Design the Web pages and Validations.
2. Develop complete UI of Front End Applications.
3. Develop complete backend Applications.
4. Develop Single page Applications using Frontend and Backend.

Mapping of Course Outcomes with Program Outcomes:

	PO												**PSO**	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	2	2	2	2	2				2			2	2	
CO2	2	2	3	2	2				3			2	2	
CO3	2	2	3	2	2				3			2	2	
CO4	2	2	3	2	2				3			2	2	
CO5	2	2	3	2	2				3			2	2	

Lab Experiments

1	Create an interactive Web page having Sign Up and Sign In forms using HTML,CSS and Bootstrap. Validate the given Sign Up and Sign In forms using JavaScript.	**CO1**
2	Introduction to Frontend (ReactJS): Installation, Project Explorer, Components. Manage state and props in React components using hooks such as useState. Use React Router to navigate between different pages of the SPA. Form Validation using React: Displaying form data to console and Web Page.	**CO2**
3	Introduction to Backend(ExpressJS): Installation, Project Explorer, connectivity. Handling Http methods using Express. Installation of axios and create an API and connecting to backend through axios.	**CO3**
4	Install JSON Server and MongoDB. Develop Single Page Application to implement CRUD operations using JSON Server or MongoDB.	**CO4**

Text Books:

1. **Frank Zammetti**,Modern Full-Stack Development Ⅱ, Apress, 2020, First Edition.
2. **Thomas A. Powell**, HTML & CSS Complete Reference,Mc Graw Hill, Fifth Edition.
3. **Thomas A. Powell**,JavaScript Complete Reference, 2012 Third Edition.

MOBILE APPLICATION DEVELOPMENT	
23CS9212	**Credits : 1**
Instruction **:**2 Periods per week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites:

- Basic programmingknowledge of Java and XML.
- Knowledge of Databases and query processing.

Course objectives:

- To facilitate students for understand and learn the basics of Android platform.
- To help students to design own applications by implementing friendly user interface.
- To get the knowledge for establishing connection to database through android applications.

Course Outcomes-At the end of the course student is able to:

CO1: Identify concepts of mobile programming that make it unique from other programming platforms.

CO2: Design user interface for mobile application using widgets with event handling

CO3: Apply layout management and Multi layout techniques to create adaptable user interface

CO4: Develop mobile applications using database connectivity for real-time applications

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	3	-	-	-	2	2	-	-	-	-	-	2	-	-
	2	3	2	2	2	3	2	-	-	-	-		2	2	
	3	3	2	2	2	3	2	-	-	-	-		3	3	
	4	3	2	2	2	3	2	-	2	2	-		3	3	

SYLLABUS

S.No	EXERCISE	Course Outcome
1	Introduction to mobile technologies and devices, Android platform and applications overview.	**CO1**
2	Installation of Android studio-Setting Android development environments	**CO1**
3	Writing Android applications -Development of Hello World Application	**CO2**
4	Develop an application that uses GUI components, Font and Colors	**CO2**
5	Develop the following application that uses Layout Managers and event listeners- Create a screen that has input boxes for User Name, Password, Address, Gender(radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (**use Linear layout)**	**CO2, CO3**
6	Design an online shopping menu application using **Grid layout.**	**CO2, CO3**
7	Design your class timetable using **Table layout**.	**CO2, CO3**
8	Write an application that draws basic graphical primitives on the screen	**CO2, CO3**
9	Design an android application to create page using Intent and one Button and pass the Values from one Activity to second Activity	**CO2, CO3**
10	Develop Registration Form application that makes use of databases	**CO4**
11	Develop a mobile application to send an e-mail	**CO4**
12	Develop a Mobile application for simple needs (Mini Project)	**CO4**

TEXT BOOKS

1. B.M. Harwani , Android Programming unleashed , Pearson, 2013.
2. Bill Phillips, Chris Stewart, Brian Hardy,Kristin Marsicano, Android Programming (Big Nerd Ranch Guide), Pearson, 2016
3. Erik Hellman, Android Programming – Pushing the limits, WILEY, 2013

REFERENCES

1. Joseph Annuzzi Jr, Lauren Darcey, Shane Condor, "Advanced Android Application Development, Developers Library", Pearson Education, 4th Edition (2015)
2. Lauren Darcey, Shane Condor, "Android, Wireless Application Development", Pearson Education, 3rd Edition.
3. Paul Deitel, Harvey Deitel, Alexander Wald, "Android 6 for programmers, An AppDriven Approach", Pearson Education
4. Rap Payne, "Beginning App Development with Flutter: Create Cross-Platform Mobile Apps", Apress (2019)

WEB REFERENCES:

1 http://developer.android.com/guide/index.html
2 http://developer.android.com/reference/packages.html
3 http://code.google.com/android/add-ons/google-apis/maps-overview.html
4 http://docs.oracle.com/javase/tutorial/

Numerical Ability & Professional Communication skills
(II Year II Sem.)

Course Category:	**Humanities**	**Credits:**	2
Branch	**All Branches**		
Course Code:	**23TP9102**	**Lecture-Tutorial-Practical:**	2+2
Prerequisites:	**Knowledge of LSRW Skills, Basic Maths**	**Continuous Evaluation:**	
		Semester End Evaluation:	
		Total Marks:	100

	Upon successful completion of the course, the student will be able to:	
Course Outcomes	**CO1**	Comprehend the essentiality of LSRW skills in paper presentations, seminars, workshops, conferences etc. with teams.**(L2),** Tosolvedifferenttypesofquestionsbasedonvocabulary,structure,grammar andverbalreasoning
	CO2	Attain the knowledge of soft skills in various conditions**(L3),** Solvequestionsbasedon sentence completion and fill in the blanks
	CO3	Explore diverse fields through English **(L4),** Tosolvedifferenttypesofquestionsbasedonvocabulary,structure,grammar andverbalreasoning
	CO4	UsetheirlogicalthinkingandanalyticalabilitiestosolveQuantitativeaptitudequestionsfromcompanyspecificandothercompetitive tests.
	CO5	Solvequestions relatedtoTime and distanceand timeandwork etc.fromcompanyspecificandother competitivetests.

Contribution of Course Outcomes towards achievement of Program Outcomes& Strength of correlations		**PO1**	**PO2**	**PO3**	**P04**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
	CO1									M	M		M		
	CO2									M	M		M		
	CO3									M	M		M		
	CO4	M													
	CO5	M													
	L- Low M-Medium H-High														

PART-A: Professional Communication skills		
UNIT-1	Abstract Preparation – Noticing Key Words –Literature Survey – Using Academi Verbs Verbal Ability : Sentence correction	**CO1**
UNIT-2	Organizational Skills – Time Management – IELTS Test Papers Exercises Verba Ability : sentence completion	**CO2**
UNIT-3	Meeting Skills – Arranging a Meeting – Prior to Meeting, During Meeting and Afte Meeting Process – Note Making – Note Taking Verbal Ability : Error Identification	**CO3**
UNIT-4	Analogy – Origin of the Words – Eponyms – MNCs Question Papers Verbal Ability : vocabulary	**CO3**

PART-B : Numerical Ability	
UNIT-I: **Numerical computation-I** Applications based on Numbers –Classification of Number System, Prime and Composite, Even and Odd Numbers, Divisibility Rule, Remainder Theorem, Finding Highest power, LCM &HCF	**CO4**
UNIT-II: Numerical estimation–I Averages, Ratio Proportion, Application of Ratios (Ages)**,**Partnerships, Sharesanddividends,	**CO4**
UNIT-III: Numerical estimation–II Percentages and its Applications, Profit Loss and Discount, Simple interest and Compound	**CO4**
UNIT-IV: Numerical estimation – II Time and work, Application of Time-work (Pipes & Cisterns), Time and Distance, circular Tracking, concept of Boats & steams.	**CO5**
UNIT-V: Numerical computation-II Mixtures and allegations, application of percentage and Ratios and Averages in Mixtures,	**CO5**

Entrepreneurship Development & IPR							
Code	Periods			Sessional Marks	End Exam Marks	Total Marks	Credits
	L	T	P				
23MC0104	30			100	-	100	-

Prerequisite:

Course Objectives: The course has been designed to develop the skills of entrepreneurship & to encourage the students to become an entrepreneur and to impart the basics of Intellectual property Rights.

Course Outcomes: At the end of the course the student will be able to:	
CO-1	Apply various theories for the entrepreneurship development ecosystem in Indian context.
CO-2	Demonstrate the ways in which entrepreneurs perceive opportunity, manage risk, organize resources and add value.
CO-3	Identify various schemes supporting entrepreneurship.
CO-4	Recognize the importance of IP and outline concepts of Intellectual Property Rights.
CO-5	Identify the significance of practice and procedure of Patents.

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO-1						1		2	1	1	2	1
CO-2						1		2	1	1	2	1
CO-3						1		2	1	1	2	1
CO-4						1		2	1	1	2	1
CO-5						1		2	1	1	2	1

Course Outcomes	PSO1	PSO2
CO-1		1
CO-2		1
CO-3		1
CO-4		1
CO-5		1

CO- Course Outcome; PO- Program Outcome; PSO-Program Specific Outcome;Level- 1: Low, 2: Medium, 3: High

SYLLABUS

UNIT - I	Periods: 4L+2T=6

UNIT TITLE: Introduction to Entrepreneurship

Entrepreneurship- Concept, Nature, Functions and Importance; Entrepreneurs Characteristics, Types and Motivation; Entrepreneurial process; Enterprise- Definition and Classification (MSME Micro, Small & Medium Enterprises).

UNIT - II	Periods: 4L+2T=6

UNIT TITLE: Entrepreneurial Journey

Creativity and Innovation, Recognizing opportunities and Generating ideas, Feasibility analysis, Industry and Competitor analysis, developing effective business model.

Class Activity:

Idea generation by students.

UNIT - III	Periods: 4L+2T=6

UNIT TITLE: Institutional Support to Entrepreneurs

Need for Institutional support different Government & Non-Government institutions to support Entrepreneurs like, NSIC, SIDO, SSIB, SSIDC, SISIs , DTICs, industrial Estates, Specialized Institutions.

UNIT - IV	Periods: 4L+2T=6

UNIT TITLE:INTRODUCTION TO IPR

Meaning of property, Origin, Nature, Meaning of Intellectual Property Rights – Introduction to TRIPS and WTO. – Kinds of Intellectual property rights—Copy Right, Patent, Trade Mark, Trade Secret and trade dress, Design, Layout Design, Geographical Indication, Plant Varieties and Traditional Knowledge.

UNIT - V	Periods: 4L+2T=6

UNIT TITLE: Patent system in India

Patents Act 1970 & Patent system in India; Patentability; Process, & product patent; filing of the patent, patent specification, patent claims, Patent opposition, & revocation, infringement, compulsory licensing, Patent Cooperation Treaty, Patent search, and patent database.

TEXT BOOKS:

1.	Robert D. Hisrich, Mathew J. Manimala, Michael P Peters and Dean A. Shepherd, "Entrepreneurship", 9th Edition, Tata Mc-graw Hill Publishing Co.ltd.-new Delhi, 2014.
2.	Bruce R. Barringer and R. Duane Ireland, "Entrepreneurship", 4th Edition, Pearson Publications, New Delhi, 2011
3.	N.K. Acharya, Text book on intellectual Property Rights, Asha Law House New Delhi, New Edition, 2001.

REFERENCE BOOKS:

1.	Narayanan, V. K., Managing technology and innovation for competitive advantage, first edition, Pearson education, New Delhi, (2006)
2.	Idris, K. (2003), Intellectual property: a power tool for economic growth, second edition, WIPO publication no. 888, Switzerland
3.	Bosworth D. & Webster E , The Management of Intellectual Property, Edward

WEB RESOURCES:

1.	https://ebooks.inflibnet.ac.in/hsp15/chapter/intellectual-property-management/
2.	https://onlinecourses.nptel.ac.in/noc20_hs66/preview