SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

DEPARTMENT OF COMPUTER SCIENCE

CBCS SINGLE MAJOR CURRICULAR FRAMEWORK
(2023-24 ADMITTED BATCH AND ONWARDS)

B.Sc. Honours (Data Science)
SEMESTER - III

S.NO	Name of the Course	Course Code	Part No	Type of the Paper	Total Marks	IA TEST	Sem End Exam	Teaching Hours			Credits
								L	T	P	C
1	DATA STRUCTURES USING C	23DSMAL231	II	Major-5	100	30	70	4	0	0	3
2	DATA STRUCTURES USING C LAB	23DSMAP231	II	Major-5 Lab	50	15	35	0	0	2	1
3	DOCUMENT ORIENTED DATABASES	23DSMAL232	II	Major-6	100	30	70	4	0	0	3
4	DOCUMENT ORIENTED DATABASES LAB	23DSMAP232	II	Major-6 Lab	50	15	35	0	0	2	1
5	Object Oriented Programming Using Java	23DSMAL233	II	Major-7	100	30	70	4	0	0	3
6	Object Oriented Programming Using Java Lab	23DSMAP233	II	Major-7 Lab	50	15	35	0	0	2	1

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				**23DSMAL231**			
Title of the Course				**Data structures using C**			
Offered to: (Programme/s)				B.Sc. Honours (Data Science)			
L	**4**	**T**	**0**	**P**	**0**	**C**	**4**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**Course Relates to:**	Local, Regional, National, Global		
Year of Revision:	**-**			**Percentage:**	**-**		
Type of the Course:				**SKILL DEVELOPMENT**			
Crosscutting Issues of the Course :				**-**			
Pre-requisites, if any							

Course Description:

This course covers fundamental concepts in data structures and algorithms. Topics include C program structure, control structures, array data structures, algorithm analysis, sorting algorithms, stacks, queues, dynamic arrays, linked lists, trees, and graphs. Students will learn implementation, properties, functions, and traversal techniques, emphasizing efficiency and practical problem-solving.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	To choose an appropriate data structure as applied to a specified problem.
2	To use various techniques for representation of the data in the real world.
3	To understand applications using data structures.
4	To develop graph data structures, traversal algorithm.
5	To test the logical ability for solving problems.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Remember C program structure, control structures, and data types.	**K1**	**PO1, PO2, PO4**	**PSO1, PSO2**
CO2	Understand arrays, Big O notation, time and space complexity.	**K2**	**PO1, PO2, PO6**	**PSO1, PSO2**
CO3	Apply sorting algorithms using arrays in practical scenarios.	**K3**	**PO2,PO3, PO4**	**PSO1, PSO2**
CO4	Analyze and compare stacks, queues, dynamic arrays, linked lists.	**K4**	**PO3,PO5, PO6**	**PSO1, PSO2**
CO5	Evaluate tree and graph data structures, traversal algorithm efficiency.	**K5**	**PO4, PO6, PO7**	**PSO1, PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**3**	**2**		**1**				**2**	**1**
CO2	**3**	**2**				**1**		**3**	**2**
CO3		**3**	**2**	**1**				**3**	**3**
CO4			**3**		**2**	**1**		**3**	**2**
CO5				**3**		**2**	**1**	**3**	**3**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure:

Unit 1: Basic Concepts and Introduction to Data Structures (12hours)

- Introduction to C program Structure
- Control Statements– IF,NESTED-IF,WHILE, DO-WHILE, FOR
- **Array Data Structure**
  - Single and Multi-Dimensional Arrays
- **Basics of Algorithm Analysis**
  - Big O Notation
  - Time and Space Complexity Notation
- **Data Structures Overview**
- **Primitive and Non-Primitive Data Structures**

Examples/Applications/Case Studies:

- Write a sample program using if, while, for, and do-while

Exercises/Projects:

- Write a C program to read an array of 10 integers and count total number of even or odd elements.
- Develop a C program to read two arrays of size 5 and store sum of these arrays into third array.

Specific Resources: (web)

1. GeeksforGeeks - Data Structures (https://www.geeksforgeeks.org/data-structures/)
2. Khan Academy - Algorithms (https://www.khanacademy.org/computing/computer-science/algorithms)

Unit 2: Sorting Algorithms (12hours)

- Implementation Using Arrays
- Bubble Sort
- Selection Sort
- Insertion Sort
- Quick Sort
- Merge Sort

Examples/Applications/Case Studies:

- Write a C program to implement the Bubble sort.
- Write a C program to implement the Insertion sort.

Exercises/Projects:

- Write a C program to implement the Selection sort .

Specific Resources: (web)

1. Geeks for Geeks - Sorting Algorithms (https://www.geeksforgeeks.org/sorting-algorithms/)
2. Sorting Algorithms (https:// www.Khanacademy. org/computing/ computerscience/ algorithms #sorting-algorithms)

Unit 3: Searching, Stack and Queue Data Structures (12hours)

- **Searching Techniques**
  - Linear Search
  - Binary Search
- **Stack Data Structures**
  - Properties and Functions
  - Applications of Stack – Infix-Postfix
- **Queue Data Structures**
  - Properties and Functions
  - Circular Queue

Examples/Applications/Case Studies:

1. Write a C program to implement the Binary Search.
2. Write a C program to implement the stack operations.

Exercises/Projects:

1. Write a C program to implement the queue operations

Specific Resources: (web)

1. GeeksforGeeks - Stack Data Structure (https://www.geeksforgeeks.org/stack-data-structure/)

2. GeeksforGeeks - Queue Data Structure (https://www.geeksforgeeks.org/queue-data-structure/)

Unit 4: Dynamic Arrays and Linked Lists (12hours)

- Introduction to Pointers
- **Dynamic Memory Allocation**
- **Linked Lists**
  - Singly Linked List
  - Doubly Linked List

Examples/Applications/Case Studies:

- Write a C program to implement Single Linked List.

Exercises/Projects:

- Write a C program to implement the Double Linked List.

Specific Resources: (web)

1. GeeksforGeeks - Linked List Data Structure (https://www.geeksforgeeks.org/data-structures/linked-list/)

2. Tutorialspoint - Dynamic Memory Allocation in C (https://www. tutorialspoint.com /cprogramming/c_dynamic_memory_allocation.htm)

Unit 5: Trees and Graphs (12hours)

- **Binary Trees**
  - Arrays and Linked Representation
  - Tree Traversals: Preorder, Inorder, Postorder
- Binary Search Tree and Operations: traversing insertion and deletion.
- **Graphs**
  - Directed and Undirected Graphs
  - Adjacency Lists and Matrices
  - Path and Graph Traversals: Breadth-First Search (BFS) and Depth-First Search (DFS)

Examples/Applications/Case Studies:

- Write a C program to implement the Binary Search Tree operations.

Exercises/Projects:

- Write a C program to implement the BFS and DFS Graph traversals.

Specific Resources: (web)

1. GeeksforGeeks - Tree Data Structure (https://www.geeksforgeeks.org/binary-tree-data-structure/)
2. GeeksforGeeks - Graph Data Structure (https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/)

TEXT BOOKS:

1. "Data Structures and Algorithms in Java" by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser , Third edition.
2. "Data Structures Using C and C++" by Yedidyah Langsam, Moshe Augenstein, Aaron M. Tenenbaum.
3. Reema Thareja, Data Structures Using C, Oxford University Press Publishers,2023.

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)
SEMESTER -END QUESTION PAPER STRUCTURE

Course Code & Title of the Course:	23DSMAL231 **& Data Structures using C**
Offered to:	B.Sc. Honours (Data Science)
Category:	**SEMESTER: 3**
Max. Marks	**70**
Max.Time	**3 Hrs**

Section A: Short Answer Questions (20 Marks)
Answer All questions. Each question carries 4 Marks.

Q1. (a)Write a C program to find the roots of a Quadratic Equation for all cases?

OR

(b) What is the difference between static and dynamic memory allocation?

Q2 (a) Write the passes to perform Selection Sort algorithm for 23,12,24,56,2,15.

OR

(b) Write an algorithm to perform Bubble Sort.

Q3 (a) Write the steps how data is stored in a Queue Data structure 21,33,12,4,56.?

OR

(b) Explain the push and pop operations in a stack data structure.

Q4 (a) What are the advantages of using a doubly linked list over a singly linked list?

OR

(b) How do insertion and deletion operations differ in linked lists?

Q5 (a) Apply preorder traversal to visit nodes of a binary tree.

OR

(b) Analyze the impact of node deletion on a binary search tree's structure.

Section B: Long Answer Questions (50 Marks)
Answer All questions. Each question carries 10 Marks.

Q6 (a)Define primitive data structures and non-primitive data structures. Give Examples.

OR

(b) Write a C program that to find the given number is prime or not.

Q7 (a) Describe the Quick Sort algorithm, including how the pivot element is chosen and how partitioning works.

OR

(b) Implement Merge Sort for given numbers 23,13,56,32,11,22,57.

Q8 (a) Define Queue Data Structure. Write the functions for enqueue and dequeue .

OR

(b) Given an infix expression, convert it to postfix notation using a stack. Explain the conversion process.

Q9 (a) Implement a C function to insert a node at the end of a singly linked list.

OR

(b) Compare the advantages of doubly linked lists over singly linked list.

Q10 (a) Analyze the results of different tree traversal methods (preorder, inorder, postorder) on a given binary tree.

OR

(b) Evaluate the strengths and weaknesses of BFS and DFS in terms of space complexity.

***

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				23DSMAP231			
Title of the Course				**Data structures using C Lab**			
Offered to: (Programme/s)				B.Sc. Honours (Data Science)			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:				**Course Relates to:**			
Year of Revision:				**Percentage:**			
Type of the Course:							
Crosscutting Issues of the Course :							
Pre-requisites, if any							

Course Description:

This course covers fundamental concepts in data structures and algorithms. Topics include C program structure, control structures, array data structures, algorithm analysis, sorting algorithms, stacks, queues, dynamic arrays, linked lists, trees, and graphs. Students will learn implementation, properties, functions, and traversal techniques, emphasizing efficiency and practical problem-solving.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	To choose an appropriate data structure as applied to a specified problem.
2	To use various techniques for representation of the data in the real world.
3	To understand applications using data structures.
4	To develop graph data structures, traversal algorithm.
5	To test the logical ability for solving problems.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Remember C program structure, control structures, and data types.	**K1**	**PO1, PO5, PO6, PO7**	**PSO1, PSO2**
CO2	Understand arrays, Big O notation, time and space complexity.	**K2**	**PO1,PO5,PO6, PO7**	**PSO1, PSO2**
CO3	Apply sorting algorithms using arrays in practical scenarios.	**K3**	**PO1,PO5, PO6, PO7**	**PSO1, PSO2**
CO4	Analyze and compare stacks, queues, dynamic arrays, linked lists.	**K4**	**PO1,PO5, PO6, PO7**	**PSO1, PSO2**
CO5	Evaluate tree and graph data structures, traversal algorithm efficiency.	**K5**	**PO1,PO5, PO6, PO7**	**PSO1, PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**-**	**2**	**2**	**2**	**2**	**1**
CO2	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**2**
CO3	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**
CO4	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**2**
CO5	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure

This lab list covers the key areas of a **Data structures using C** course, providing hands-on practice with (Turbo C++)

Unit 1: [Introduction to Data Structures and Basic Concepts](6Hrs)

Lab 1:

1. Write a C program to find Compound Interest.
2. Write a C Program to find Roots of a quadratic equations with nature of roots.
3. Write a C program to find factorial of a given number.

- **Dataset** (web link) / **Experiment**: Explore basic programming constructs in C.
- **Tasks:**

1. **Basic Program:** Write a C program to calculate Simple Interest.
2. **Control Structures:** Write a C program using if, while, for, and do-while loops.

Lab 2:

1. Write a C program to read n numbers and count number of +ve,-ve, zeros.
2. Write a C program to find sum of digits of a given number.
3. Write a C program to find Armstrong numbers between 1-1000.

- **Dataset (web link) / Experiment:** Perform basic operations on using while,for in C.
- **Tasks:**

1. **Count Even/Odd Elements:** Write a C program to count even and odd elements of 10 integers.
2. **Strong Number:** Write a C program to find given number is strong or not.

Lab 3:

1. Write a C program to read elements and display in matrix.
2. Write a C program to display the identity matrix of given size.

3. Write a C program to find multiplication of two matrices.

- **Dataset (web link) / Experiment:** Perform basic operations on using while, for in C.
- **Tasks:**

1. Transpose of Matrix: Write a C program to transpose a matrix.
2. Sum of matrices: Write a C program to find sum of two matrices.

Unit 2: [Sorting Algorithms](6Hrs)

Lab 4:

1. Write a C program to implement the Bubble sort.
2. Write a C program to implement the Insertion sort.

- **Dataset** (web link) / **Experiment**: Implement and understand basic sorting algorithms.
- **Tasks:**

1. **Bubble Sort:** Write a C program to sort an array using Bubble Sort.
2. **Insertion Sort:** Write a C program to sort an array using Insertion Sort.

Lab 5:

1. Write a C program to implement the Selection sort.
2. Write a C program to implement the Quick sort.

- **Dataset (web link) / Experiment:** Implement advanced sorting algorithms.
- **Tasks:**

1. **Selection Sort:** Write a C program to sort an array using Selection Sort.
2. **Quick Sort:** Write a C program to sort an array using Quick Sort.

Unit 3: [Stack and Queue Data Structures](6Hrs)

Lab 6:

1. Write a C program to implement the stack operations.

- **Dataset** (web link) / **Experiment**: Implement basic stack operations (`push`, `pop`, `and peek`) to understand the Last-In-First-Out (LIFO) principle in C.
- **Tasks:**

1. **Stack Operations:** Write a C program to perform push, pop, display operations on a stack.

Lab 7:

1. Write a C program to implement the queue operations

- **Dataset (web link) / Experiment:** Implement basic queue operations (`enqueue`, `dequeue`) to understand the First-In-First-Out (FIFO) principle in C.
- **Tasks:**

1. **Queue Operations:** Write a C program to perform enqueue, dequeue, and display operations on a queue.

Unit 4: [Dynamic Arrays and Linked Lists](6Hrs)

Lab 8:

1. Write a C program to implement the infix to postfix expression

- **Dataset** (web link) / **Experiment**: Convert infix expressions to postfix in C.
- **Tasks:**

1. **Infix to Postfix Conversion:**

Write a C program to convert infix expressions to postfix.

Unit 5: [Trees and Graphs] (6Hrs)

Lab 9:

1. Write a C program to implement the Binary Search Tree operations.

- **Dataset** (web link) / **Experiment**:Implement Binary Search Tree (BST) operations to manage hierarchical data in C.
- **Tasks:**

BST Operations: Write a C program to perform insert, delete, and search operations on a Binary Search Tree (BST).

Lab 10:

1. Write a C program to implement the BFS traversal algorithm for a graph.

2. Write a C program to implement the DFS traversal algorithm for a graph.

- **Dataset** (web link) / **Experiment**:Implement BFS and DFS traversal algorithms for exploring graphs in C.
- **Tasks:**

1. **BFS Traversal:** Write a C program to perform Breadth-First Search (BFS) on a graph.

2. **DFS Traversal:**Write a C program to perform Depth-First Search (DFS) on a graph.

Lab Manual:

1. "Data Structures and Algorithms in Java" by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser , Third edition.

References:

1. "Data Structures Using C and C++" by Yedidyah Langsam, Moshe Augenstein, Aaron M. Tenenbaum.
2. Reema Thareja, Data Structures Using C, Oxford University Press Publishers,2023.

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				23DSMAL232			
Title of the Course				**Document Oriented Databases**			
Offered to: (Programme/s)				B.Sc. Honours (Data Science)			
L	**4**	**T**	**0**	**P**	**0**	**C**	**4**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**Course Relates to:**	Local, Regional, National, Global		
Year of Revision:	**-**			**Percentage:**	**-**		
Type of the Course:				**SKILL DEVELOPMENT**			
Crosscutting Issues of the Course :				**-**			
Pre-requisites, if any				**Basic of Excel and Data handling**			

Course Description:

This course provides a comprehensive overview of database management systems, focusing on both traditional SQL databases and modern NoSQL solutions like MongoDB. Starting with foundational concepts, it covers the roles and advantages of DBMS, data models, and SQL commands for database creation and manipulation. Students will explore entity relationships, normalization processes, and SQL constraints. The course then transitions to NoSQL databases, specifically MongoDB, addressing its features, installation, and operations such as CRUD, indexing, and sharding. The final unit emphasizes data modeling and aggregation techniques in both SQL and MongoDB, equipping students with skills to design and manage robust databases effectively.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	To educate student regarding databases and how to manage databases.
2	To handle the large amount of data handling demands of business.
3	To implement a data store that provides high performance, high availability, and automatic scaling
4	To Process an immense diversity of data that needs to be stored and processed.
5	To make use of features and functionalities to work on NO SQL Data Base Mongo DB

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Recap of the data handling about database. DBMS Architecture	**K1**	**PO3 PO5 PO6**	**-**
CO2	Understand databases, various features of Database , the installation procedure of Oracle.	**K2**	**PO3 PO5 PO6**	**PSO2**
CO3	Apply Query on Mongo DB's rich query language to support Create, Read, Update, and Delete (CRUD) operations.	**K3**	**PO3 PO5 PO6**	**PSO1 PSO2**
CO4	Analyze the aggregation framework to perform aggregation operations.	**K4**	**PO3 PO5 PO6**	**PSO1 PSO2**
CO5	Evaluate the performance of both SQL and No Sql Databases.	**K5**	**PO3 PO5 PO6**	**PSO1 PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	-	-	2	-	2	2	-	-	-
CO2	-	-	2	-	3	3	-	-	1
CO3	-	-	2	-	3	3	-	2	1
CO4	-	-	2	-	3	3	-	2	1
CO5	-	-	2	-	3	3	-	2	1

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure:

Unit – 1 : Overview of Database Management Systems: (12Hrs)

Introduction to Data, information, data vs. information –database and DBMS Role and advantages of DBMS – types of databases –problems with file system data management.

Data Models:

The importance of Data models –The evolution of Data Models-Degrees of data abstraction

Introduction to Sql-Data Definition Commands – Data Types - Creating Table Structures - advanced data definition commands - alter – drop

Examples/Applications/Case Studies:

- Library database for book and member management.
- Customer relationship management (CRM) system for tracking customer interactions.

Exercises/Projects:

- Design a database for a small retail store.
- Implement a student information system using SQL.

Specific Resources: (web)

W3Schools SQL Tutorial

Unit –2 :Entity Types and SQL constraints: (12Hrs)

Entity Super types and Subtypes- entity integrity - selecting primary keys - Natural Keys and Primary Keys - The need for normalization – The normalization process – converting to first normal form – conversion to second normal form – conversion to third normal form – higher level normal forms.

SQL Constraints Adding Table Rows Saving Table Changes - Updating Table Rows - Restoring Table Contents - Deleting Table Rows

Examples/Applications/Case Studies:

- Employee database with Person as supertype, Employee and Manager as subtypes.
- Employee database with Person as supertype, Employee and Manager as subtypes.

Exercises/Projects:

- Design a normalized school administration database with entity integrity.
- Implement entity integrity and constraints in a hospital management system using SQL.

Specific Resources: (web)

W3Schools SQL Constraints

Unit – 3 :Data Manipulation Language: (12Hrs)

Select Queries: Selecting Rows with Conditional Restrictions – operators - advanced select queries – SQL functions

Mongo DB Features and Installation, The Need for No SQL Databases, What Are No SQL Databases? CAP Theorem, BASE Approach, Types of NoSQL Databases, MongoDB Features, Document Database MongoDB Is Schemaless MongoDB Uses BSON , Rich Query Language, Aggregation Framework Indexing, GridFS, Replication, Sharding The mongo Shell , Terms Used in MongoDB, Data Types in MongoDB, Working with Database Commands, Create Database, Drop Database , Display List of Databases, Display the Version of MongoDB, Display a List of Commands.

Examples/Applications/Case Studies:

- Installing configuring running of Mongo db

- Working with data base commands in mongo db

Exercises/Projects:
- Working with crud operations in mongo db

Specific Resources: (web)
MongoDB Official Documentation

Unit – 4 :Mongo DB Operations: (12Hrs)
MongoDB CRUD Operations,Collections, Create a Collection, Create Capped Collections, Create Operations, Insert Documents, Read Operations, Query Documents, Update Operations, Update Documents, Delete Operations, Delete Documents, Working with Arrays, Working with Arrays, Query for Null or Missing Fields, To Query Null or Missing Fields, Working with the limit() and skip() Methods, limit() and skip() Methods.

Examples/Applications/Case Studies:
- Implementing aggregation operations in mongo db
- Implementing index operations

Exercises/Projects:
- Working with replication in mongo db
- Working with sharding in mongo db.

Specific Resources: (web)
MongoDB Documentation

Unit – 5 : [Data Modelling and Aggregation] (12Hrs)
Data Modelling and Aggregation, Data Models, Embedded Data Models, Normalized Data Models
Data Model Relationship Between Documents, Data Model Using an Embedded Document, Data Model Using Document References, SQL Aggregation Terms and Corresponding MongoDB, Aggregation Operators, Matching SQL Aggregation to MongoDB, Aggregation Operations.

Examples/Applications/Case Studies:
- Working with multi document transactions in mongo db

Exercises/Projects:
- Working with multi document transactions with conflicts in mongo db

Specific Resources: (web)
MongoDB Aggregation Documentation

Text Books:
1. “Fundamentals of Database Systems” by R.Elmasri and S.Navathe, Pearson Education; Seventh edition
2. “Database System Concepts” by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill, Seventh edition
3. MongoDB Recipes: With Data Modeling and Query Building Strategies By Subhashini Chellappan, Dharanitharan Ganesan , Publisher : Apress

***

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)
SEMESTER -END QUESTION PAPER STRUCTURE

Course Code & Title of the Course:	23DSMAL232 **Document Oriented Databases**
Offered to:	B.Sc. Honours (Data Science)
Category: MAJOR	**SEMESTER: 3**
Max. Marks	**70**
Max.Time	**3 Hrs**

Section A: Short Answer Questions (20 Marks)

Answer All questions. Each question carries 4 Marks.

Q1 (a) Explain the Three-Level Architecture?

OR

(b) Explain the advantages of DBMS approach?

Q2 (a) Explain the fundamental operations in Relational algebra?

OR

(b) Discuss about union, intersect and minus in SQL?

Q3 (a) List any five data types in MongoDB?

OR

(b) Write Create, Drop, List Database in MongoDB.

Q4 (a) Define CRUD . Write about MongoDB Query Language

OR

(b) What are different types of NoSQL?

Q5 (a) How does MongoDB Aggregation differ from SQL aggregation?

OR

(b) List and explain four MongoDB aggregation operations.

Section B: Long Answer Questions (50 Marks).

Answer All questions. Each question carries 10 Marks.

Q6 (a) Explain the ER diagrams indetail and explain the notations which are used in ER diagrams?

OR

(b) Describe the structure of DBMS with Neat Diagram? Compare and contrast file systems with database Systems?

Q7 (a) Define normalization? Explain briefly about 1NF,2NF, 3NF BCNF,4NF and 5NF with suitable examples?

OR

(b) Explain the role of functional dependencies in normalization with suitable Example?

Q8 (a) Explain in detail about CAP Theorem.

OR

(b) Define MongoDB.Is MongoDBSchemaless, Explain with an example.

Q9 (a) Define CRUD . Explain CRUD Operations in detail.

OR

(b) What is an Array? How it is implemented in MongoDB.

Q10 (a) Discuss the advantages and challenges of Data Model Relationships in MongoDB. Include examples.

OR

(b) Compare and contrast Embedded and Normalized Data Models in MongoDB with real-world examples.

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				23DSMAP232			
Title of the Course				**Document Oriented Databases Lab**			
Offered to: (Programme/s)				B.Sc. Honours (Data Science)			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**Course Relates to:**			Local, Regional, National, Global
Year of Revision:				**Percentage:**			
Type of the Course:				**SKILL DEVELOPMENT**			
Crosscutting Issues of the Course :							
Pre-requisites, if any				**Basic of Excel and Data handling**			

Course Description:

This course provides a comprehensive overview of database management systems, focusing on both traditional SQL databases and modern NoSQL solutions like MongoDB. Starting with foundational concepts, it covers the roles and advantages of DBMS, data models, and SQL commands for database creation and manipulation. Students will explore entity relationships, normalization processes, and SQL constraints. The course then transitions to NoSQL databases, specifically MongoDB, addressing its features, installation, and operations such as CRUD, indexing, and sharding. The final unit emphasizes data modeling and aggregation techniques in both SQL and MongoDB, equipping students with skills to design and manage robust databases effectively.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	To educate student regarding databases and how to manage databases.
2	To handle the large amount of data handling demands of business.
3	To implement a data store that provides high performance, high availability, and automatic scaling
4	To Process an immense diversity of data that needs to be stored and processed.
5	To make use of features and functionalities to work on NO SQL Data Base Mongo DB

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	**Recap** of the data handling about database. DBMS Architecture	**K1**	**PO3 PO5 PO6**	**-**
CO2	Understand databases, various features of Database, the installation procedure of Oracle.	**K2**	**PO3 PO5 PO6**	**PSO2**
CO3	Apply Query on Mongo DB's rich query language to support Create, Read, Update, and Delete (CRUD) operations.	**K3**	**PO3 PO5 PO6**	**PSO1 PSO2**
CO4	Analyze the aggregation framework to perform aggregation operations.	**K4**	**PO3 PO5 PO6**	**PSO1 PSO2**
CO5	Evaluate the performance of both SQL and No Sql Databases.	**K5**	**PO3 PO5 PO6**	**PSO1 PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**-**	**-**	**2**	**-**	**2**	**2**	**-**	**-**	**-**
CO2	**-**	**-**	**2**	**-**	**3**	**3**	**-**	**-**	**1**
CO3	**-**	**-**	**2**	**-**	**3**	**3**	**-**	**2**	**1**
CO4	**-**	**-**	**2**	**-**	**3**	**3**	**-**	**2**	**1**
CO5	**-**	**-**	**2**	**-**	**3**	**3**	**-**	**2**	**1**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

This lab list covers the key areas of a ….(title of the course) course, providing hands-on practice with ….(technology/software)

Unit 1: [**Overview of Database Management Systems**] (6Hrs)

Lab 1:

- Create a student information system using SQL.
- **Dataset** (web link) / **Experiment**: Sample SQL database schema for student records.
- **Tasks:**

Design and implement a SQL-based student information system to manage student records, including enrollment, grades, and personal information.

Lab 2:

- Retrieve student information by executing SQL SELECT queries, filtering by conditions like student ID or name.
- **Dataset** (web link) / **Experiment**: SQL student database (provided schema or sample data).
- **Tasks:**

1. Query the student information system using SQL SELECT commands, filtering results by specific student attributes.

Unit 2: [**Entity Types and SQL constraints**] (6Hrs)

Lab 3:

- Installing configuring running of Mongo db
- **Dataset** (web link) / **Experiment**: MongoDB Installation Guide (official MongoDB documentation).
- **Tasks:**

Install, configure, and run MongoDB on various operating systems.

Lab 4:

- Working with data base commands in mongo db
- **Dataset** (web link) / **Experiment**: MongoDB Command Reference (MongoDB official documentation).
- **Tasks:**
- Practice key database commands in MongoDB, including querying, updating, and managing collections.

Unit 3: [**Data Manipulation Language**] (6Hrs)

Lab 5:

- Working with crud operations in mongo db
- **Dataset** (web link) / **Experiment**: MongoDB CRUD Operations Tutorial (official MongoDB documentation).
- **Tasks:**

Perform Create, Read, Update, and Delete operations on MongoDB collections using practical examples.

Lab 6:

- Implementing aggregation operation s in mongo db
- **Dataset** (web link) / **Experiment**: MongoDB Aggregation Pipeline Guide (official MongoDB documentation).
- **Tasks:** Use MongoDB's aggregation framework to perform complex data processing, filtering, grouping, and summarization.

Unit 4: [**Mongo DB Operations**] (6Hrs)

Lab 7:

- Implementing index operations
- **Dataset** (web link) / **Experiment**: MongoDB Indexing Guide (official MongoDB documentation).
- **Tasks:** Create and manage indexes to optimize query performance, including single-field, compound, and text indexes.

Lab 8:

1. Working with replication in mongo db
2. Working with sharding in mongo db.

- **Dataset** (web link) / **Experiment**: MongoDB Replication Setup Guide (official MongoDB documentation) and MongoDB Sharding Documentation (official MongoDB guide).
- **Tasks:**

1. Set up a MongoDB replica set to explore data replication, failover procedures, and read/write distribution.
2. Set up sharding to horizontally scale a MongoDB database, managing data distribution and handling query routing.

Unit 5: [**Data Modelling and Aggregation**] (6Hrs)

Lab 9:

- Working with multi document transactions in mongo db
- **Dataset** (web link) / **Experiment**: MongoDB Transactions Documentation (official MongoDB guide).
- **Tasks:** Implement multi-document transactions to ensure atomicity across multiple operations, maintaining data consistency.

Lab 10:

- Working with multi document transactions with conflicts in mongo db
- **Dataset** (web link) / **Experiment**: MongoDB Conflict Resolution Guide (official MongoDB documentation).
- **Tasks:** Manage and resolve conflicts in multi-document transactions, applying conflict resolution techniques to maintain consistency.

Lab Manual:

1. "Fundamentals of Database Systems" by R.Elmasri and S.Navathe, Pearson Education; Seventh edition.

References:

1. "Database System Concepts" by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill, Seventh edition
2. MongoDB Recipes: With Data Modeling and Query Building Strategies By Subhashini Chellappan, DharanitharanGanesan ,Publisher : Apress

***

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				**23DSMAL233**			
Title of the Course				**Object Oriented Programming Using Java**			
Offered to: (Programme/s)				**B. Sc. Honours (Data Science)**			
L	**4**	**T**	**0**	**P**	**0**	**C**	**4**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**MAJOR**			**Course Relates to:**	**GLOBAL**		
Year of Introduction:	**2024**			**Percentage:**	**NA**		
Type of the Course:				**Employability**			
Crosscutting Issues of the Course :				**-**			
Pre-requisites, if any				**Programming Knowledge**			

Course Description:
This course on Object-Oriented Programming using Java introduces fundamental concepts and techniques essential for Java programming. Students will explore the object-oriented paradigm, basic Java syntax, and program structure, covering variables, data types, operators, and control statements. The course delves into classes, objects, methods, inheritance, arrays, and interfaces. It further includes multi-threading, exception handling, and Java packages. Students will gain practical skills in graphics programming using the Graphics class and file handling, including reading and writing files. By the end of the course, students will be adept at designing and implementing Java applications with robust functionalities.

Course Aims & Objectives:

S. No	COURSE OBJECTIVES
1	Learn the basics of Java programming and Object-Oriented Programming (OOP) concepts.
2	Understand and use Java control statements, classes, objects, and methods to build structured programs.
3	Implement inheritance, arrays, and interfaces to create efficient and reusable Java code.
4	Learn how to handle errors and manage multiple threads to improve program performance.
5	Evaluate file handling and create simple graphics in Java to solve practical programming problems.

Course Outcomes:
At the end of the course, the student will / will be…

NO	COURSE OUTCOME	BTL	PO	PSO
CO1	**Remember** the basic concepts of Object-Oriented Programming (OOP) and the structure of a simple Java program.	**K1**	**PO1, PO5, PO6, PO7**	**PSO1, PSO2**
CO2	**Explain** how control statements and classes work in Java to manage the flow and structure of a program.	**K2**	**PO5, PO6, PO7**	**PSO1, PSO2**
CO3	**Apply** concepts of inheritance, arrays, and interfaces to create basic Java programs.	**K3**	**PO5, PO6, PO7**	**PSO1, PSO2**
CO4	**Analyze** how multi-threading and exception handling can be used to improve program performance and error management.	**K4**	**PO5, PO6, PO7**	**PSO1, PSO2**

CO5	**Evaluate** file handling techniques in Java and **Create** simple graphics programs using the Java Graphics class.	**K5, K6**	**PO5, PO6, PO7**	**PSO1, PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO-PSO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**-**	**2**	**3**	**2**	**3**	**3**
CO2	**-**	**-**	**-**	**-**	**3**	**3**	**2**	**3**	**3**
CO3	**-**	**-**	**-**	**-**	**3**	**3**	**2**	**3**	**3**
CO4	**-**	**-**	**-**	**-**	**3**	**3**	**2**	**3**	**3**
CO5	**-**	**-**	**-**	**-**	**3**	**3**	**2**	**3**	**3**

Use the codes 3, 2, 1 for High, Moderate and Low correlation Between CO-PO-PSO respectively.

Course Structure:

Unit – I: Introduction to Java Programming (12 Hrs.)

Introduction-Object Oriented paradigm-Basic Concepts of OOP-Benefits of OOP-Applications of OOP- Java features-Simple Java program structure-Java tokens-Java Statements-Implementing a Java Program-Java Virtual Machine-Command line arguments-Constants-Variables-Data Types-Declaration of Variables-Giving Value to Variables-Scope of variables-Symbolic Constants-Type casting-Getting Value of Variables - types of operators with examples-expressions

Description:

This course is tailored to understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.

Examples:

1. Operators concept in java
2. Type casting in java

Exercises:

1. Design Java program to perform Type Casting in java.
2. Develop a Java program for sorting a given list of names in ascending order.

Learning Outcomes:

By the end of the unit, students will understand the concept and underlying principles of Object-Oriented Programming and object-oriented concepts are incorporated into the Java programming language

Web Resources:

- Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur."Basic Concepts of Java Programming", 2018.

https://www.youtube.com/watch?v=OjdT2l-EZJA&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=1

Unit – II: Control statements, Classes, Objects and Methods (12 Hrs.)

Introduction-Decision making with if statement-Simple if statement-If Else statement-Nesting of if else statements-The else if ladder-The switch statement-The conditional operator-The While statement-The do-while statement-The for statement- Jumps in loops-Defining a class-Adding variables-Adding methods-Creating objects-Accessing class members-Constructors-Method overloading-Static members-Nesting of methods

Description:

This unit provides fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.

Examples:

1. Control statements in java
2. Constructors, Method overloading, Static keyword in java

Exercises:

1. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
2. Construct a Java program that implements method overloading

Learning Outcomes:

By the end of this unit, students will be able to gain knowledge in Implementing Object Oriented Programming Concepts like class, constructor, overloading concepts in java

Web Resources:

- Introduction to Classes and Objects in Java ,Neso Academy, 7 june 2020

https://www.youtube.com/watch?v=W-D71ZeMixQ&list=PLBlnK6fEyqRiwWLbSXKFtdGV8OVqr9dZr

Unit – III: Inheritance, Arrays, Strings and Interfaces (12 Hrs.)

Extending a class-Overloading methods-Final variables and methods-Final classes-Abstract methods and classes-Arrays-One dimensional arrays- Creating an array – Two dimensional arrays- Strings- Wrapper classes.

MULTIPLE INHERITANCE: Introduction- Defining interfaces- Extending interfaces-Implementing interfaces-Accessing interface variables.

Description:

This unit helps in understanding the principles of inheritance and interfaces, array creation and string handling functions

Examples:

1. Types of inheritances.
2. String handling functions and array creation in java

Exercises:

1.

Learning Outcomes:

By the end of this unit, students will be able to understand and implement inheritance and interfaces, array creation and string handling functions in a Java program.

Web Resources:

1. Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur.“Inheritance in Java”, 2018.

https://www.youtube.com/watch?v=rxsl1TzcEgg

2. Arrays in Java by Neso Academy, 2019

https://www.youtube.com/watch?v=kWJHzambtNo&list=PLBlnK6fEyqRiraym3T703apTvEZLaSVtJ

3. What is string in Java by Lab Mug, 2023

https://www.youtube.com/watch?v=Vv8ijzbz22s

Unit – IV: Multi-Threading, Exception Handling and Packages (12 Hrs.)

Introduction-Creating Threads-Extending the Threads-Stopping and Blocking a Thread-Lifecycle of a Thread-Using Thread Methods-Thread Exceptions-Thread Priority-Implementing the ‘Runnable’ Interface-Types of errors-Compile time errors-Run-time errors-Exceptions-Exception handling-Multiple Catch Statements-Using finally statement-Java API Packages-Creating Packages-Accessing a Package- Using a Package.

Description:

This unit helps in understanding and implementing multi-threaded programs, Exception handling and packages.

Examples:

1. Multi-threading in java
2. Types of exception handling mechanisms

Exercises:

1. **Construct Java program to implement various types of Exception Handling Mechanisms**
2. **Design a program to create and Import Packages**

Learning Outcomes:

By the end of this unit, students will be able to Implement Multithreading, exception handling and packages in Java

Resources:

1. Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur."Packages in Java", 2018. https://www.youtube.com/watch?v=TwU3cv1FFis&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=17
2. Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur."Exception Handling in Java", 2018. https://www.youtube.com/watch?v=vUov8EkjZjU&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=23
3. Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur."Multi Threading in Java", 2018. https://www.youtube.com/watch?v=6rYOyIGfy3w&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=27

Unit – V: Graphics Programming and I/O Files (12 Hrs.)

Graphics Programming: The Graphics class Lines and rectangles-Circles and ellipses-Drawing arcs -Line graphs -Drawing Bar charts.

I/O files: Concept of Streams-Stream classes-Byte Stream Classes-Character Stream classes: Reader stream classes, Writer Stream classes-Reading and writing files.

Description:

This unit focuses Understanding the principles of applets, I/O streams in java and java database connectivity

Examples:

- Writing and Reading Files.

Exercises:

- Develop a program for writing and reading Files
- Draw a circle with a radius of 50 pixels.

Learning Outcomes:

By the end of this unit, students will be able to implement graphical user interface in Java programs, Input/output Streams in java and java database connectivity with oracle

Resources:

- "File Handling in Java", Learn Coding, 2021. https://www.youtube.com/watch?v=VJgCjLuU4e8&list=PLqleLpAMfxGDVu5tUmUg9jSQUUB8_5DB0

Specific Resources:

Text Books:

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.

Reference Books:

1. Programming in Java by Sachin Malhotra, OXFORD University Press
2. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TATA McGraw-Hill Company.
3. Deitel&Deitel. Java TM: How to Program, PHI (2007)
4. Java Programming: From Problem Analysis to Program Design- D.S Mallik
5. Object Oriented Programming Through Java by P. Radha Krishna, Universities Press (2008)

Web Resources:

1. Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur."Basic Concepts of Java Programming", 2018. https://www.youtube.com/watch?v=OjdT2l-EZJA&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=1

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)
SEMESTER -END QUESTION PAPER

Course Code & Title of the Course:	**23DSMAL233 Object Oriented Programming Using Java**
Offered to:	**B. Sc. Honours (Data Science)**
Category:	**SEMESTER: 3**
Max. Marks	**70**
Max.Time	**3 Hrs**

Section A

ANSWER THE FOLLOWING QUESTIONS. 5 X 4 = 20 Marks

1. (a) Explain Java virtual Machine.

OR

(b) Explain Variables in java with examples.

2. (a) Develop a java code to demonstrate labelled loops.

OR

(b) Develop a java code to print the factorial of a given number.

3. (a) Illustrate vectors with an example.

OR

(b) Illustrate multithread with an example.

4. (a) How do you draw a rectangle using the Graphics class? Write a basic example.

OR

(b) Develop a Java program for drawing Bar charts.

5. (a) Explain the need of Files in handling data.

OR

(b)Write about reading and writing files with examples.

Section B

ANSWER THE FOLLOWING QUESTIONS. 5 X 10 = 50 Marks

6. (a) Summarise History and Features of Java.

OR

(b) Summarise Operators and expressions in java with examples.

7. (a) Illustrate string methods in java with examples.

OR

(b) Illustrate accessing single dimensional arrays in Java with example.

8. (a) Explain thread life cycle.

OR

(b) Explain packages in java with an example.

9. (a) Explain the methods you used to draw shapes. Write a Java program to draw a circle and a rectangle using the Graphics.

OR

(b) Explain Drawing Lines and arcs in java with an example.

10. (a) Explain streams in java with example.

OR

(b) Explain I/O classes in Java with examples.

@@@@@@@@@@@@@@@

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				**23DSMAP233**			
Title of the Course				**Object Oriented Programming using Java Lab**			
Offered to: (Programme/s)				B.Sc. Honours (Data Science)			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:				**Course Relates to:**		**GLOBAL**	
Year of Revision:				**Percentage:**			
Type of the Course:				**Employability**			
Crosscutting Issues of the Course :				**-**			
Pre-requisites, if any				**Programming knowledge**			

Course Description:
This lab course focuses on practical Java programming skills. Students will explore type casting, sorting, and object-oriented concepts such as class creation, method overloading, and constructors. They will work on matrix multiplication, inheritance types, and runtime polymorphism. The course includes multi-threading, exception handling, and package management, providing hands-on experience with error handling and concurrent programming. Additionally, students will learn to create applets and manage file operations, including reading and writing files. This course offers a thorough understanding of Java programming through diverse, real-world applications and challenges.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	**Learn** to use basic Java programming to handle type casting, sorting, and data operations.
2	**Create** Java classes and methods, including constructors, to perform calculations and manage data.
3	**Develop** programs for matrix operations, different types of inheritance, and interfaces.
4	**Understand** and use multi-threading and exception handling in Java, and work with packages.
5	**Practice** creating applets and managing file operations, including reading and writing files.

Course Outcomes
At the end of the course, the student will be able to…

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	**Apply** type casting and sorting techniques in Java to process and organize data effectively.	**K3**	**PO2, PO5, PO6**	**PSO1,PSO2**
CO2	**Design** and **implement** classes, methods, and constructors to perform calculations and manage object attributes in Java.	**K6**	**PO2, PO3, PO5, PO6**	**PSO1,PSO2**
CO3	**Develop** Java programs to demonstrate various types of inheritance, matrix operations, and interface implementations.	**K6**	**PO2, PO3, PO5, PO6**	**PSO1,PSO2**
CO4	**Analyze** and **handle** multi-threading and exception scenarios, including creating threads, managing exceptions, and demonstrating package usage.	**K4**	**PO2, PO3, PO5, PO6**	**PSO2**

CO5	**Construct** and **manage** applets and file operations, including writing and reading files, to create interactive applications and handle file data.	**K6**	**PO2, PO3, PO5, PO6**	**PSO1,PSO2**

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**-**	**2**	**2**	**-**	**2**	**1**
CO2	**2**	**-**	**2**	**-**	**2**	**3**	**-**	**2**	**1**
CO3	**-**	**-**	**2**	**-**	**3**	**3**	**2**	**2**	**3**
CO4	**-**	**-**	**2**	**-**	**3**	**3**	**3**	**-**	**1**
CO5	**-**	**-**	**2**	**-**	**2**	**3**	**-**	**2**	**1**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure

This lab list covers the key areas of Object Oriented Programming in Java Lab course, providing hands-on practice

Unit-1: Introduction to Java Programming (6 Hrs)

Lab 1

1. Design Java program to perform Type Casting in java.

- **Dataset (web link) / Experiment:** Java Type Casting Example and
- **Tasks:** Demonstrate type casting with different data types, including integer to double and string to integer conversions.

2. Develop a Java program for sorting a given list of names in ascending order.

- **Dataset (web link) / Experiment:** Java Sorting Example
- **Tasks:** Sort a list of names in ascending order using Java's built-in sorting methods.

Unit-2: Control statements, Classes, Objects and Methods (6 Hrs)

Lab 2

3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.

- **Dataset (web link) / Experiment:** Java Rectangle Class Example
- **Tasks:** Define Rectangle class with methods to calculate perimeter, area, and read attributes (length and width) from user.

4. Construct a Java program that implements method overloading.

- **Dataset (web link) / Experiment:** Java Method Overloading Example
- **Tasks:** Implement method overloading by defining multiple methods with the same name but different parameters for various operations.

Lab 3

5. Write a program to demonstrate Parameterized Constructors.

- **Dataset (web link) / Experiment:** Explore constructor usage in Java.
- **Tasks:** Create a class with a parameterized constructor to initialize attributes with given values and demonstrate its application.

6. Write a Program to demonstrate Constructor Overloading.

- **Dataset (web link) / Experiment:** Understand constructor overloading in Java.
- **Tasks:** Implement multiple constructors within a class, each with different parameter lists to illustrate how constructor overloading works.

Unit-3: Inheritance, Arrays, Strings and Interfaces (6 Hrs)

Lab 4

7. Design a Java program to calculate multiplication of 2 matrices.

- **Dataset (web link) / Experiment:** Explore matrix operations in Java.
- **Tasks:** Implement a program to perform matrix multiplication using nested loops and display the resultant matrix.

8. Develop a Java program for extending and implementing interfaces.

- **Dataset (web link) / Experiment:** Understand interfaces in Java.
- **Tasks:** Create an interface and a class that implements this interface. Extend the class to demonstrate interface inheritance and implementation.

Lab 5

9. Construct Java programs to implement various types of inheritance

i. Single ii. Multi-Level iii. Hierarchical iv. Hybrid

- **Dataset (web link) / Experiment:** Explore different inheritance types in Java.
- **Tasks:** Implement Java programs to demonstrate single, multi-level, hierarchical, and hybrid inheritance using appropriate class structures.

10. Develop a java program to implement runtime polymorphism.

- **Dataset (web link) / Experiment:** Learn about runtime polymorphism in Java.
- **Tasks:** Create a program demonstrating runtime polymorphism by using method overriding and dynamic method dispatch with base and derived classes.

Lab 6

11. Write a program to demonstrate Abstract Classes and Final Keyword

- **Dataset (web link) / Experiment:** Understand abstract classes and final keyword in Java.
- **Tasks:** Create a Java program using abstract classes and methods, and apply the final keyword to classes and methods to prevent inheritance and modification.

12. Design a program for extending and implementing interfaces.

- **Dataset (web link) / Experiment:** Explore interfaces in Java.
- **Tasks:** Develop a Java program that defines and implements multiple interfaces, demonstrating how to extend and use interface methods in a class.

Unit-4: Multi-Threading, Exception Handling and Packages (6 Hrs)

Lab 7

13. Write a Java program which accepts withdraw amount from the user and throws an exception "In Sufficient Funds" when withdraw amount more than available amount.

- **Dataset (web link) / Experiment:** Explore exception handling in Java.
- **Tasks:** Write a Java program to accept a withdrawal amount from the user and throw an "Insufficient Funds" exception if the withdrawal amount exceeds the available balance.

14. Construct a Java program to create three threads and that displays "good morning", for every one second, "hello" for every 2 seconds and "welcome" for every 3 seconds by using extending Thread class.

- **Dataset (web link) / Experiment:** Explore multi-threading in Java.
- **Tasks:** Create a Java program with three threads that display "good morning" every 1 second, "hello" every 2 seconds, and "welcome" every 3 seconds by extending the Thread class.

Lab 8

15. Construct a Java program that creates three threads. First thread displays "OOPS", the second thread displays "Through" and the third thread Displays "JAVA" by using Runnable interface.

- **Dataset (web link) / Experiment:** Explore thread creation using the Runnable interface.
- **Tasks:** Write a Java program to create three threads where the first thread displays "OOPS", the second thread displays "Through", and the third thread displays "JAVA" using the Runnable interface.

16. Design a program to create and Import Packages

- **Dataset (web link) / Experiment:** Explore Java package creation and import.
- **Tasks:** Create a Java program to define and import custom packages.

Lab 9

17. Construct Java program to implement various types of Exception Handling Mechanisms

i. Arithmetic Exception

ii. Number Format Exception
iii. ArrayIndexOutofBounds Exception

- **Dataset (web link) / Experiment:** Explore different exception handling mechanisms.
- **Tasks:** Construct a Java program to demonstrate handling:
- Arithmetic Exception
- Number Format Exception
- ArrayIndexOutOfBounds Exception

18. Design a program to demonstrate Catch Blocks

- **Dataset (web link) / Experiment:** Explore the usage of catch blocks in exception handling.
- **Tasks:** Design a Java program to demonstrate the use of catch blocks for handling exceptions.

Unit-5: Streams, Files, Graphic Programming (6 Hrs)

Lab -10

19. Develop a program for writing and reading Files

- **Dataset (web link) / Experiment:** Explore file I/O operations.
- **Tasks:** Develop a Java program to write data to a file and read it back.

20. Write a java program to Draw a circle with a radius of 50 pixels.

- **Dataset (web link) / Experiment:** Explore graphics programming with Java.
- **Tasks:** Write a Java program to draw a circle with a radius of 50 pixels using the Graphics class.

References:

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.

Web Resources:

- Prof.DebasisSamanta, Dept of Computer science, IIT Kharagpur."Basic Concepts of Java Programming", 2018.

https://www.youtube.com/watch?v=OjdT2l-EZJA&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=1

***

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				**23DSMIT231**			
Title of the Course				**Data Analysis Using Python**			
Offered to: (Programme/s)				**B.Sc. Honours (Statistics/ Computer Science Sec-B)**			
L	**4**	**T**	**0**	**P**	**0**	**C**	**4**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:				**Course Relates to:**		**Global**	
Year of Revision:				**Percentage:**			
Type of the Course:				**Skill Development**			
Crosscutting Issues of the Course :				**-**			
Pre-requisites, if any							

Course Description:

This course offers a detailed introduction to Data Science, emphasizing practical applications in Python. It covers key areas such as data exploration, cleaning, and visualization using libraries like NumPy, pandas, and matplotlib. Students will learn to handle and analyze data with pandas, create and manipulate arrays with NumPy, and visualize data with matplotlib and seaborn. The course also includes advanced topics such as data aggregation, group operations, and time series analysis, equipping students with the skills to manage and interpret complex datasets effectively. Ideal for those seeking a solid foundation in data science with hands-on Python experience.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Understand the basics of Data Science, including the data analysis process, and how to use Python tools like iPython and Jupyter Notebook for data analysis.
2	Apply NumPy to handle arrays and matrices, including creating, reshaping, and performing operations on them.
3	Use pandas to analyze data by cleaning, organizing, and exploring datasets, and create visualizations to gain insights.
4	Perform data wrangling by combining and reshaping datasets, and use visualization tools like matplotlib and seaborn to present data effectively.
5	Conduct advanced data analysis by aggregating and grouping data, and analyze time series data with techniques such as resampling and frequency conversion.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	**Understand** the basics of Data Science, the data analysis process, and how Python tools like iPython and Jupyter Notebook can help.	K2	PO1, PO5, PO6, PO7	PSO1, PSO2
CO2	**Apply** NumPy to work with arrays and matrices, including creating, reshaping, and performing basic operations on them.	K3	PO1, PO5, PO6, PO7	PSO1, PSO1
CO3	**Analyze** data using pandas by cleaning, organizing, and exploring datasets, and create simple visualizations to understand the data better.	K4	PO1, PO5, PO6, PO7	PSO1, PSO1
CO4	**Evaluate** and combine data from different sources, reshape it, and use visualization tools like matplotlib and seaborn to gain insights.	K5	PO1, PO5, PO6, PO7	PSO1, PSO1
CO5	**Create** advanced data analysis techniques by working with grouped data, time series, and more complex operations to uncover deeper insights.	K6	PO1, PO5, PO6, PO7	PSO1, PSO1

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**-**	**3**	**2**	**3**	**3**	**2**
CO2	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**2**
CO3	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**
CO4	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**
CO5	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure:

Unit – 1 : [Introduction to Python and Its Libraries] (12Hrs)

Introduction: Introduction to Data Science, Exploratory Data Analysis and Data Science Process. Motivation for using Python for Data Analysis, Introduction of Python shell iPython and Jupyter Notebook.

Essential Python Libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels.

Examples/Applications/Case Studies:

- Write a Program to print "Hello World!" using Jupyter Notebook.
- Write a program to access the elements in a list using Jupyter Notebook.

Exercises/Projects:

- Write a program toDisplay Keys and values in a Dictionary using Jupyter Notebook.

Specific Resources: (web)

- W3Schools

Unit – 2 : [Numpy] (12Hrs)

Numpy: NumPy Arrays - difference between python lists and NumPy array, What is NumPy array, creating basic array, adding, removing and sorting elements, reshaping array, converting 1d array to 2d array, indexing and slicing, creating array from existing data, creating matrices, getting random numbers getting count and unique numbers, transposing and reshaping a matrix, reverse an array, reshaping multidimensional arrays.

Examples/Applications/Case Studies:

- Create a panda's series from a dictionary of values and andarray.
- Give an example to create a DataFrame from a singlendarray.

Exercises/Projects:

- Write a Pandas program to select the rows where thescore is missing, i.e. NaN.
- Write a program to generate a series of float numbersfrom 21.0 to 30.0 with an increment of 1.5 each.

Specific Resources: (web)

- W3Schools

Unit – 3 : [Pandas] (12Hrs)

Pandas: Introduction, Getting Started, Series, Data Frame**,** Read CSV, Read JSON -Analyzing DataFrames, Cleaning Data, Cleaning Empty Cell, Cleaning Wrong Format, Cleaning Wrong Data,Removing Duplicates, Correlations, Plotting.

Examples/Applications/Case Studies:

- Write a program to generate a series of the first 10 numbers.
- Write a Pandas program to count the number of rows and columns of a Data Frame.

Exercises/Projects:

- Write a program to generate a series and print the top 3 elements using the headfunction.
- Write a program in Python to create a Series in Python from the given dictionary.

D= {"Jan" : 31, "Feb" : 28, "Mar" : 31}.

Specific Resources: (web)

- W3Schools

Unit – 4: [Data Wrangling and Data Visualization] (12Hrs)

Data Wrangling: Hierarchical Indexing, Combining and Merging Data Sets Reshaping and Pivoting. **Data Visualization matplotlib**: Basics of matplotlib, plotting with pandas and seaborn, other python visualization tools.

Examples/Applications/Case Studies:

- Plotting a line chart of date versus temperature by adding Label on X and Y axis, and adding a Title and Grids to the chart.
- Design a Plotting Histogram

Exercises/Projects:

- To plot a bar chart, we will specify kind='bar'. We can alsospecify the DataFrame columns to be used as x and y axes. Let us now add a column "Days" consisting of day names to"MelaSales.csv" .
- Use Matplotlib and Seaborn to create line charts, bar charts, and scatter plots from a given dataset. Focus on different types of visualizations to represent data effectively.

Specific Resources: (web)

- Matplotlib Documentation
- Seaborn Documentation
- Data Wrangling with Pandas

Unit – 5 : [Data Aggregation and Time Series Data Analysis] (12Hrs)

Data Aggregation and Group operations: Group by Mechanics, Data aggregation, General split-apply-combine, Pivot tables and cross tabulation.

Time Series Data Analysis: Date and Time Data Types and Tools, Time series Basics, date Ranges, Frequencies and Shifting, Time Zone Handling, Periods and Periods Arithmetic, Resampling and Frequency conversion, Moving Window Functions.

Examples/Applications/Case Studies:

- How does the "pd.Series" function utilize the dates variable to create the time series data.
- What is the purpose of using the groupby function? How does it group the data and what is the result of applying the sum function to the grouped data?

Exercises/Projects:

- What does the pivot_table function do in this code? Explain how it transforms the DataFrame df and the purpose of the index, columns, and aggfunc parameters.

SpecificResources: (web)

- Data Aggregation Guide
- Time Series Analysis Tutorial
- Matplotlib Basics

Text Books/References:

- McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media
- O'Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline O'Reilly Media

***

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)
SEMESTER -END QUESTION PAPER STRUCTURE

Course Code & Title of the Course:	**23DSMIT231** **Data Analysis Using Python**
Offered to:	**B.Sc. Honours (Statistics/ Computer Science Sec-B)**
Category:	**SEMESTER: 3**
Max. Marks	**70**
Max.Time	**3 Hrs**

Section A: Short Answer Questions (20 Marks)

Answer All questions. Each question carries 4 Marks.

Q1 (a) Explain the motivation for using Python in Data Analysis.

OR

(b) Describe the steps in the Data Science process.

Q2 (a) How do you create a basic NumPy array? Provide an example.

OR

(b) What is the difference between Python lists and NumPy arrays? Explain with an example.

Q3 (a) How can you read a CSV file into a Pandas DataFrame?

OR

(b)Explain how to clean empty cells in a Pandas DataFrame.

Q4 (a) What is Hierarchical Indexing in Pandas? Explain its importance.

OR

(b) Describe how to plot data using Matplotlib.

Q5 (a) What is the purpose of the groupby function in Pandas?

OR

(b) Explain the basics of Time Series data analysis in Python.

Section B: Long Answer Questions (50 Marks)

Answer All questions. Each question carries 10 Marks.

Q6 (a) How is Python used in Data Science? Give examples of Jupyter Notebook features.

OR

(b) What are the main steps in the Data Science process? How do Python libraries help?

Q7 (a) What is the difference between Python lists and NumPy arrays? Show examples.

OR

(b) How do you perform basic operations like reshaping with NumPy arrays?

Q8 (a) How do you create and clean a DataFrame in Pandas? Explain with a CSV example.

OR

(b) How can you manage missing values and duplicates in Pandas?

Q9 (a) What is Hierarchical Indexing in Pandas? How do you use Matplotlib for plotting?

OR

(b) How do you merge and reshape data in Pandas? Give examples.

Q10 (a) How does the groupby function work in Pandas? Provide an example.

OR

(b) What are the key techniques for analyzing time series data in Python?

Note:

- Short answer questions assess foundational knowledge (Remembering, Understanding and Apply).
- This structure emphasizes a focus on higher-order thinking skills (Understand, Application, Analysis, and Evaluation) in the long answer section.
- Consider including a mix of question types within each section to ensure a comprehensive assessment.

SRI DURGA MALLESWARA SIDDHARTHA MAHILA KALASALA:: VIJAYAWADA
(An autonomous college in the jurisdiction of Krishna University, Machilipatnam)

Course Code				**23DSMIL231**			
Title of the Course				**Data Analysis Using Python Lab**			
Offered to: (Programme/s)				**B.Sc. Honours (Statistics/ Computer Science Section-B)**			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Minor**			**Course Relates to:**	**Global**		
Year of Revision:	**-**			**Percentage:**			
Type of the Course:				**Skill Development**			
Crosscutting Issues of the Course :				**-**			
Pre-requisites, if any							

Course Description:

This course offers a detailed introduction to Data Science, emphasizing practical applications in Python. It covers key areas such as data exploration, cleaning, and visualization using libraries like NumPy, pandas, and matplotlib. Students will learn to handle and analyze data with pandas, create and manipulate arrays with NumPy, and visualize data with matplotlib and seaborn. The course also includes advanced topics such as data aggregation, group operations, and time series analysis, equipping students with the skills to manage and interpret complex datasets effectively. Ideal for those seeking a solid foundation in data science with hands-on Python experience.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Understand the basics of Data Science, including the data analysis process, and how to use Python tools like iPython and Jupyter Notebook for data analysis.
2	Apply NumPy to handle arrays and matrices, including creating, reshaping, and performing operations on them.
3	Use pandas to analyze data by cleaning, organizing, and exploring datasets, and create visualizations to gain insights.
4	Perform data wrangling by combining and reshaping datasets, and use visualization tools like matplotlib and seaborn to present data effectively.
5	Conduct advanced data analysis by aggregating and grouping data, and analyze time series data with techniques such as resampling and frequency conversion.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO 1	**Understand** the basics of Data Science, the data analysis process, and how Python tools like iPython and Jupyter Notebook can help.	K 2	PO1, PO5, PO6, PO7	PSO1, PSO2
CO 2	**Apply** NumPy to work with arrays and matrices, including creating, reshaping, and performing basic operations on them.	K 3	PO1, PO5, PO6, PO7	PSO1, PSO1
CO 3	**Analyze** data using pandas by cleaning, organizing, and exploring datasets, and create simple visualizations to understand the data better.	K 4	PO1, PO5, PO6, PO7	PSO1, PSO1
CO 4	**Evaluate** and combine data from different sources, reshape it, and use visualization tools like matplotlib and seaborn to gain insights.	K 5	PO1, PO5, PO6, PO7	PSO1, PSO1
CO 5	**Create** advanced data analysis techniques by working with grouped data, time series, and more complex operations to uncover deeper insights.	K 6	PO1, PO5, PO6, PO7	PSO1, PSO1

For BTL: K1: Remember; K2: Understand; K3: Apply; K4: Analyze; K5: Evaluate; K6: Create

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1	**1**	**-**	**-**	**-**	**3**	**2**	**3**	**3**	**2**
CO2	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**2**
CO3	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**
CO4	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**
CO5	**1**	**-**	**-**	**-**	**3**	**3**	**3**	**3**	**3**

Use the codes 3,2,1 for High, Moderate and Low correlation Between CO-PO-PSO respectively

Course Structure

This lab list covers the key areas of a ….(title of the course) course, providing hands-on practice with ….(technology/software)

Unit 1: [**Introduction to Python and Its Libraries**] (6Hrs)

Lab 1:

1. Write a Program to print "Hello World!" using Jupyter Notebook.
2. Write a program to access the elements in a list using Jupyter Notebook.

- **Dataset** (web link) / **Experiment**: Learn basic operations in Python using Jupyter Notebook.
- **Tasks:**

1. **Hello World Program:**Write a Python program to print "Hello, World!" in Jupyter Notebook.
2. **List Element Access:** Write a Python program to access elements in a list using Jupyter Notebook.

Lab 2:

1. Write a program toDisplay Keys and values in a Dictionary using Jupyter Notebook.

- **Dataset**(web link) / **Experiment**: Explore data structures in Python using Jupyter Notebook.
- **Tasks:**

1. **Dictionary Keys and Values:**

Write a Python program to display keys and values in a dictionary using Jupyter Notebook.

Unit 2: [**Numpy**] (6Hrs)

Lab 3:

1. Create a panda's series from a dictionary of values and a ndarray.
2. Give an example to create a DataFrame from a single ndarray.

- **Dataset** (web link) / **Experiment**: Utilize pandas to handle data structures in Python.
- **Tasks:**

1. **Create pandas Series:**

Write a Python program to create a pandas Series from a dictionary and anndarray.

2. **Create DataFrame:**

Write a Python program to create a DataFrame from a single ndarray using pandas.

Lab 4:

1. Write a Pandas program to select the rows where the score is missing, i.e. NaN.
2. Write a program to generate a series of float numbers from 21.0 to 30.0 with an increment of 1.5 each.

- **Dataset**(web link) / **Experiment**: Practice data manipulation and series creation in pandas.
- **Tasks:**

1. **Select Rows with NaN:**

Write a pandas program to select rows where the score is missing (NaN).

2. **Generate Float Series:**

Write a Python program to generate a series of float numbers from 21.0 to 30.0 with a 1.5 increment.

Unit 3: [**Pandas**] (6Hrs)

Lab 5:

1. Write a program to generate a series of the first 10 numbers.
2. Write a Pandas program to count the number of rows and columns of a Data Frame.

- **Dataset** (web link) / **Experiment**: Explore basic operations in pandas with series and DataFrames.
- **Tasks:**

1. **Generate Number Series:**

Write a Python program to generate a series of the first 10 numbers.

2. **Count Rows and Columns:**

Write a pandas program to count the number of rows and columns in a DataFrame.

Lab 6:

1. Write a program to generate a series and print the top 3 elements using the head function.
2. Write a program in Python to create a Series in Python from the given dictionary.

D= {“Jan” : 31, “Feb” : 28, “Mar” : 31}.

- **Dataset**(web link) / **Experiment**: Perform basic operations with pandas Series.
- **Tasks:**

1. **Top 3 Elements with head():**

Write a Python program to generate a series and print the top 3 elements using the head() function.

2. **Create Series from Dictionary:**

Write a Python program to create a Series from the dictionary D = {"Jan": 31, "Feb": 28, "Mar": 31}.

Unit 4: [**Data Wrangling and Data Visualization**] (6Hrs)

Lab 7:

1. Plotting a line chart of date versus temperature by adding Label on X and Y axis, and adding a Title and Grids to the chart.
2. Design a program to visualize data with a histogram.

- **Dataset** (web link) / **Experiment**: Create visual representations of data using matplotlib.
- **Tasks:**

1. **Line Chart:**

Plot a line chart of date versus temperature, adding labels on the X and Y axis, a title, and grids.

2. **Histogram Visualization:**

Design a program to visualize data using a histogram.

Lab 8:

1. To plot a bar chart, we will specify kind=’bar’. We can also specify the DataFrame columns to be used as x and y axes. Let us now add a column “Days” consisting of day names to “MelaSales.csv”.
2. Use Matplotlib and Seaborn to create line charts, bar charts, and scatter plots from a given dataset. Focus on different types of visualizations to represent data effectively.

- **Dataset**(web link) / **Experiment**: Enhance data visualization skills using Matplotlib and Seaborn.
- **Tasks:**

1. **Bar Chart with Days Column:**

Plot a bar chart specifying kind='bar', adding a "Days" column to MelaSales.csv for the X-axis.

2. **Multiple Visualizations:**

Use Matplotlib and Seaborn to create line charts, bar charts, and scatter plots, focusing on effective data representation.

Unit 5: [**Data Aggregation and Time Series Data Analysis**] (6Hrs)

Lab 9:

1. How does the “pd.Series” function utilize the dates variable to create the time series data.
2. What is the purpose of using the groupby function? How does it group the data and what is the result of applying the sum function to the grouped data?

- **Dataset** (web link) / **Experiment**:Explore time series creation and data grouping in pandas.
- **Tasks:**

1. **Time Series with pd.Series:**

Utilize the dates variable with pd.Series to create time series data.

2. **Groupby and Sum Function:**

Understand the purpose of the groupby function, how it groups data, and the result of applying the sum function to grouped data.

Lab 10:

1. What does the pivot_table function do in this code? Explain how it transforms the DataFrame df and the purpose of the index, columns, and aggfunc parameters.

- **Dataset** (web link) / **Experiment**:Learn to use pivot tables for data transformation in pandas.
- **Tasks:**

1. **Pivot Table Function:**

Explain the pivot_table function in pandas, detailing how it transforms DataFramedf using the index, columns, and aggfunc parameters.

Lab Manual:

- McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media

References:

1. McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media
2. O'Neil, C., & Schutt, R. (2013). Doing Data Science: Straight Talk from the Frontline O'Reilly Media

***