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

B.Sc. Honours (Artificial Intelligence)
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	23AIMAL231	II	Major-5	100	30	70	4	0	0	3
2	DATA STRUCTURES USING C LAB	23AIMAP231	II	Major-5 Lab	50	15	35	0	0	2	1
3	DOCUMENT ORIENTED DATABASES	23AIMAL232	II	Major-6	100	30	70	4	0	0	3
4	DOCUMENT ORIENTED DATABASES LAB	23AIMAP232	II	Major-6 Lab	50	15	35	0	0	2	1
5	Object Oriented Programming Using Java	23AIMAL233	II	Major-7	100	30	70	4	0	0	3
6	Object Oriented Programming Using Java Lab	23AIMAP233	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				23AIMAL231			
Title of the Course				**Data structures using C**			
Offered to: (Programme/s)				**B.Sc. Honours (Artificial Intelligence)**			
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:	23AIMAL231 **Data Structures using C**
Offered to:	B.Sc. Honours (Artificial Intelligence)
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				23AIMAP231			
Title of the Course				**Data structures using C Lab**			
Offered to: (Programme/s)				B.Sc. Honours (Artificial Intelligence)			
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				23AIMAL232			
Title of the Course				**Document Oriented Databases**			
Offered to: (Programme/s)				B.Sc. Honours (Artificial Intelligence)			
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:	23AIMAL232 **Document Oriented Databases**
Offered to:	B.Sc. Honours (Artificial Intelligence)
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				23AIMAP232			
Title of the Course				**Document Oriented Databases Lab**			
Offered to: (Programme/s)				B.Sc. Honours (Artificial Intelligence)			
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

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: [**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				**23AIMAL233**			
Title of the Course				**Object Oriented Programming Using Java**			
Offered to: (Programme/s)				**B. Sc. Honours (Artificial Intelligence)**			
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:	**Object Oriented Programming Using Java**
Offered to:	**23AIMAL233**
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				**23AIMAP233**			
Title of the Course				**Object Oriented Programming using Java Lab**			
Offered to: (Programme/s)				B.Sc. Honours (Artificial Intelligence)			
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 (6Hrs)

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 (6Hrs)

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 (6Hrs)

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 (6Hrs)

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 (6Hrs)

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				**23AIMIL231**			
Title of the Course				**Data Analysis Using Python**			
Offered to: (Programme/s)				**B.Sc. Honours (Computer Science with Cognitive System), B.Sc. Honours Computer Science Sec-A**			
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
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:

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 andprint the top 3 elements using the headfunction.
- Write a program in Python to create aSeries 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:	**23AIMIL231** **Data Analysis Using Python**
Offered to:	**B.Sc. Honours (Computer Science with Cognitive System), B.Sc. Honours (Computer Science) Sec-A**
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				**23AIMIP231**			
Title of the Course				**Data Analysis Using Python Lab**			
Offered to: (Programme/s)				**B.Sc. Honours (Computer Science with Cognitive System), B.Sc. Honours Computer Science Sec-A**			
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

***

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

Course Code				23BCMAL231			
Title of the Course				**Data Structures**			
Offered to: (Programme/s)				**BCA Honours**			
L	**4**	**T**	**0**	**P**	**2**	**C**	**4**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**Course Relates to:**	**Global**		
Year of Revision:				**Percentage:**			
Type of the Course:				**Employability**			
Crosscutting Issues of the Course :							
Pre-requisites, if any				Basic knowledge of programming concepts Familiarity with the C programming language is recommended.			

Course Description:

To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Understand various Data Structures for data storage and processing.
2	Realize Linked List Data Structure for various operations
3	Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures.
4	Understand and implement various searching & sorting techniques
5	Understand the Non-Linear Data Structures such as Binary Trees and Graphs

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Describe and differentiate between various data structures and their uses.	**K2**	**6,7**	**1,2**
CO2	Implement and manipulate data structures using C.	**K3**	**6,7**	**1,2**
CO3	Analyze and evaluate the efficiency of algorithms.	**K4**	**6,7**	**1,2**
CO4	Solve complex problems by selecting and applying appropriate data structures.	**K3**	**6,7**	**1,2**
CO5	Demonstrate proficiency in dynamic memory management and pointer manipulation in C.	**K3**	**6,7**	**1,2**

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**	**3**		
CO2						**3**	**2**		
CO3					**3**	**2**	**3**		
CO4						**3**	**3**		
CO5						**3**	**3**		

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

Course Structure:

Data structures

Unit -1: Introduction to data structures: Types of data structures-Primitive data structures, Nonprimitive data structures – linear data structures, nonlinear data structures, real world applications of data structures, Abstract data types-ADT for stack, queue, linked list, Performance analysis of algorithms-time complexity, space complexity. (10Hrs)

Description:

Data structures are fundamental concepts in computer science and programming, designed to organize, manage, and store data efficiently. Understanding data structures is essential for solving complex problems and optimizing the performance of software.

Examples:

Time Complexity: Looking up a specific page number in a well-organized notebook. If you know the page number, you can go directly to that page without flipping through the rest of the notebook. The time taken is the same regardless of how many pages are in the notebook.

Space Complexity: Exchanging two items between your hands. No matter how large the items or how many times you swap, you only need a fixed amount of space (your two hands). Similarly, the algorithm only requires a constant amount of extra space, regardless of the input size.

Exercises

Program to insert, update, delete an element

Learning Outcomes:

Understand various Data Structures for data storage and processing.

Specific Resources: (web)

https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Unit – 2 : Linear Data Structures **(**14Hrs)

Linked List: Introduction to Linked Lists, linked lists ADT, Comparison between Linked List and Array, Types of Linked Lists and their implementations - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Application of linked lists

Description:

Linear data structures are data structures where elements are arranged sequentially, one after another. In a linear data structure, each element has a unique predecessor and successor (except the first and last elements). These structures are simple and easy to implement, making them foundational in computer science.

Examples:

The university's administration requires a system to manage student records, which include operations such as adding, searching, updating, and deleting student records as well as deleting student reports

Exercises:

Implement Single Linked List with insertion, deletion and traversal operations

Learning Outcomes:

Realize Linked List Data Structure for various operations

Specific Resources: (web)

https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

Unit – 3 : Stacks: (14Hrs)

Introduction to stack, Stack ADT, stacks using array and Linked List, Application of stacks –Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation - Tower of Hanoi, Recursion: Introduction to Queue, Queue ADT, Queues using arrays and Linked List, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue

Description:

A stack is a linear data structure that follows the Last In, First Out (LIFO) principle. This means that the last element added to the stack will be the first one to be removed. Stacks are used in various applications, including algorithm implementation, memory management, and backtracking problems.

Examples:

To store data of books in a last-in, first-out (LIFO) manner.

An online bookstore needs to manage its inventory, process customer orders, and recommend books to users. To achieve these tasks efficiently, the bookstore must use various data structures.

Exercises:
Programs to implement the Queue operations using an array and linked Lists
Learning Outcomes:
Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures
Specific Resources: (web)
https://onlinecourses.swayam2.ac.in/nou24_cs15/preview
Unit – 4 : Searching: (8 Hrs)
Linear or Sequential Search, Binary Search and Indexed Sequential Search Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort
Description:
Searching is the process of finding a particular element or a set of elements in a collection of data. It is a fundamental operation in computer science, crucial for various applications like databases, information retrieval, and algorithms
Examples:To search books based on user requirement such as specific book title , author etc …. Imagine an online bookstore where books are stored in an array or a list. If a user searches for a book by its title, the system can use linear search to find the book.
Exercises:
- program to search an item in a given list using Linear Search & Binary Search.
- Searching Algorithms
- program for implementation of Bubble Sort Insertion Sort Quick Sort Sorting Algorithms

Learning Outcomes:
Understand and implement various searching & sorting techniques.
Specific Resources: (web)
https://onlinecourses.swayam2.ac.in/nou24_cs15/preview
Unit – 5 : Binary Trees: (14Hrs)
Introduction to Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree. Graphs: Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs (DFS, BFS), Application of Graphs.
Description:
A binary tree is a hierarchical data structure in which each node has at most two children, referred to as the left child and the right child. Binary trees are used in various applications, such as searching, sorting, and representing hierarchical data like file systems.
Examples:
To search books based on user requirement such as ISBN or ISSN number. Imagine an online bookstore where books are stored in an array or a list. If a user searches for a book by its ISSN or ISBN number, the system can use binary search tree to store and retrieve the book based on unique keys.
Exercises:
program for Binary Search Tree Traversals
Learning Outcomes:
Understand the Non-Linear Data Structures such as Binary Trees and Graphs
Specific Resources: (web)
https://onlinecourses.swayam2.ac.in/nou24_cs15/preview
Text Books:
1. Horowitz and Sahani, “Fundamentals of Data Structures”, Galgotia Publications Pvt Ltd Delhi India.
2. A.K. Sharma ,Data Structure Using C, Pearson Education India.
3. “Data Structures Using C” Balagurusamy E. TMH

Reference Books
1. “Data Structures through C”, Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, “Data Structure Using C and C++” Wiley Dreamtech Publication. 3. Lipschutz, “Data Structures” Schaum’s Outline Series, Tata Mcgraw-hill Education (India)Pvt. Ltd .

4. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.

***

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:	**23BCMAL231 DATASTRUCTURES**
Offered to:	**BCA Honours**
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.

1 a) Write ADT for stacks and explain it. K1
(or)
b) Write real world applications of data structures K1
2 a) compare linked lists with arrays K2
(or)
b) Explain about different types of linked lists K2
3 a) Write differences between stacks and queues K2
(or)
b) convert following expression from infix to postfix. K2
a+b*c+(d*e+f) +g.
4 a) Write program for linear search. K1
(or)
b) Write program for Bubble sort. K1
5 a) Discuss applications of graphs. K2
(or)
b) Explain with examples sequential and linked representation of graphs. K2

Long Answer Questions (50 Marks)

Section B:

Answer All questions. Each question carries 10 Marks.

6 a) Give classification of Data structures and explain them. K2
(or)
b) Explain about analysis of algorithms. K2
7 a) Develop code insertion and deletion in single linked list. K2
(or)
b) Write functions for insertion, display of values in doubly linked list. K2
8 a) Write code to implement queues using arrays. K2
(or)
b) write code to implement stacks using linked list. K2
9 a) Write program for binary search. K2
(or)
b) Apply quick sort for below given values and write code to implement quick sort. 11 2 9 13 57 25 17 1 90 3. K2
10 a) Explain with code deletion in binary search tree. K2
(or)
b) Explain Depth first search with an example. K2

****

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

Course Code				**23BCMAP231**			
Title of the Course				**DATA STRUCTURES LAB**			
Offered to: (Programme/s)				**BCA Honours**			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:		**2024-25**		**Semester:**			**3**
Course Category:		**MAJOR**		**Course Relates to:**		**GLOBAL**	
Year of Revision:				**Percentage:**			
Type of the Course:				**Major**			
Crosscutting Issues of the Course:							
Pre-requisites, if any				**Programming knowledge**			

Course Description:

The objective of course is to provide students with practical experience in using data structures. Students will learn to perform data manipulation and retrieval, implement advanced techniques in real life applications.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Students will learn to implement fundamental data structures such as arrays, linked lists, stacks, queues, and hash tables.
2	Students will explore and implement more complex data structures including trees and graphs.
3	Students will analyse the time and space complexity of different data structures and their operations.
4	Students will apply data structures to solve practical problems, enhancing their problem-solving and programming skills.
5	Students will improve their proficiency in programming languages commonly used for data structures, such as C++, Java, or Python.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Implement and manipulate basic and advanced data structures.	**K2**	**6,7**	**1,2**
CO2	Analyze the performance of data structures and algorithms.	**K3**	**6,7**	**1,2**
CO3	Apply data structures to solve practical computing problems.	**K3**	**6,7**	**1,2**
CO4	Develop efficient and optimized code for various data structure operations.	**K3**	**6,7**	**1,2**
CO5	Demonstrate proficiency in a programming language used for data structure implementation.	**K3**	**6,7**	**1,2**

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**	**3**	**2**	**2**
CO2						**3**	**2**	**2**	**3**
CO3					**3**	**2**	**3**	**3**	**2**
CO4						**3**	**3**	**2**	**3**
CO5						**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 lab course, providing hands-on practice with various data structures, enabling students to implement and manipulate these structures to solve real-world problems efficiently. Through a series of programming exercises and projects, students will develop practical skills in designing, analysing, and optimizing data structures.

Unit 1: Basic Concepts: (6Hrs)

Lab 1:

1. Time Complexity calculation on Linear & Binary Search

Exercise 1:

Linear and binary search :

Objective: Learn to calculate time complexity on linear search binary search

Tasks:

Write a program to calculate time complexity for

- **Linear Search**
- **Binary Search**

Lab 2:

2. **Time Complexity calculation on Bubble Sort**

Bubble Sort

Objective: Learn to calculate time complexity on Bubble sort

Tasks:

Write a program to calculate time complexity for Bubble Sort

Unit 2: Linear Data Structures

Lab 3:

1. Single Linked Lists

Representation of Single Linked Lists

1. Objective: To understand the concept and types of linked lists better

2. Tasks:

Write Program to implement Single Linked List with insertion, deletion and traversal operations

Lab 4:

2. Double Linked Lists

Representation of Double Linked Lists

1. Objective: To understand the concept and types of linked lists better

2. Tasks:

Write Program to implement Double Linked List with insertion, deletion and traversal operations

Lab 4:

3. Circular Linked Lists

Representation of Circular Linked Lists

1. Objective: To understand the concept and types of linked lists better

2. Tasks:

3. Write Program to implement Circular Linked List with insertion, deletion and traversal operations

Unit 3: Stacks

Lab 5:

1. Stack Operations

Implementing stacks in linked Lists and arrays

1. **Objective:** Understanding to implement stacks in linked lists and arrays

Lab 6:

Write Programs to implement the Stack operations using an array **.**

Write Programs to implement the Stack operations using Linked List.

Write Programs to implement the Queue operations using an array.

Write Programs to implement the Queue operations using Linked List.

Unit 4:

Searching

1. Quick Sort

Lab 7:

Objective: Implementation of Sorting Algorithms

1. Tasks:

a.) Write a program for implementation of the following Sorting Algorithms i)Bubble Sort ii) Insertion Sort iii)Quick Sort

b.) Write a program for implementation of Selection Sort

Unit 5: Binary Search Trees (6Hrs)

Lab 8:

Creation of binary trees and tree traversals

1. Binary Trees:

Objective: Understanding **Creation of binary trees and tree traversals**

Write a program for Binary Search Tree Traversals

References:

1. "Data Structures through C", Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata Mcgraw-hill Education (India)Pvt. Ltd .
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.

***

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

23BCMAP231: DATA STRUCTURES LAB

Offered to: BCA Honours **Semester: III**
Max. Marks: 50 (CIA: 15 + SEE: 35) **Hrs/Week: 2**

Model Paper: Practicals

Time: 3 Hrs. **Max. Marks: 35**

Section – A

1. Experiment-1 15 M
2. Experiment-2 10 M

Section – B

Viva Voce 10 M

@@@@

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

Course Code				**23BCMAL232**			
Title of the Course				**Database Management Systems**			
Offered to: (Programme/s)				**BCA Honours**			
L	**4**	**T**	**0**	**P**	**0**	**C**	**4**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**Course Relates to:**	**Global**		
Year of Revision:				**Percentage:**			
Type of the Course:				Employability			
Crosscutting Issues of the Course:							
Pre-requisites, if any				**Knowledge in data structures**			

Course Description:

This course provides a comprehensive introduction to the principles and practices of database management systems. Students will start with fundamental concepts, including database users, characteristics, and advantages of the DBMS approach. They will learn about data models, schemas, and database architectures. As they progress, students will explore data modeling using the ER model, the relational data model, and SQL for database operations. Advanced topics include normalization, relational database design, transaction processing, and concurrency control techniques. By the end of this course, students will have a solid understanding of designing, managing, and optimizing databases efficiently.

Course Aims and Objectives:

S. NO	COURSE OBJECTIVES
1	Introduce students to the fundamental concepts of DATABASE.
2	Explain the architecture and components of database systems, including data models, schemas, instances, and the three-schema architecture
3	Solve real-world database design problems by applying normalization techniques and understanding functional dependencies to ensure data integrity and efficiency
4	Demonstrate the process of data modeling using the Entity-Relationship (ER) model and relational model, emphasizing the importance of attributes, keys, and constraints.
5	Familiarize students with SQL and PL/SQL including schema definition, constraints, queries, and views, to proficiently interact with and manipulate relational databases.

Course Outcomes

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Understand fundamental database concepts and architecture and data models	K2	**6,7**	**1,2**
CO2	Normalize schemas to ensure data integrity and reduce redundancy	**K3**	**6,7**	**1,2**
CO3	Demonstrate proficiency in using SQL for defining and manipulating database structures	**K4**	**6,7**	**1,2**
CO4	Develop the ability to perform data retrieval using joins, subqueries and nested subqueries	**K3**	**6,7**	**1,2**
CO5	Gain proficiency in developing PL/SQL programs and handling exceptions.	**K4**	**6,7**	**1,2**

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**	**3**	**2**	**2**
CO2						**3**	**2**	**2**	**3**
CO3					**3**	**2**	**3**	**3**	**2**
CO4						**3**	**3**	**2**	**3**
CO5						**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 : Overview of Database Systems (12Hrs)

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications.

Data Models: Introduction; types of data models, Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Description:

A database system is a system for managing data that allows users to store, modify, and extract information from a database. It provides a systematic and organized way of managing data.

Learning Outcome:

- Understand the fundamental concepts and purpose of database systems.
- Develop the ability to differentiate between databases and traditional file systems.
- Appreciate the role of databases in modern applications and enterprises.
- **Exercises:**

Databases and Database Users

- Explain with an example how a database differs from traditional file processing systems.
- Discuss the advantages of using a DBMS approach over traditional file processing systems.
- Outline the characteristics that define the database approach.

Database System Concepts and Architecture

- Differentiate between data models, schemas, and instances, using examples where applicable.
- Explain the concept of three-schema architecture and how it achieves data independence.
- Compare and contrast centralized and client/server architectures for DBMSs.
- Classify different types of database management systems based on their characteristics and functionalities.
- **Specific Resources:**
- Fundamentals ofDatabase System, Esraa Adnan Hadi.

https://www.researchgate.net/publication/336472480_Fundamentals_of_Database_System

- Fundamentals of Database Systems Fourth Edition, Ramez Elmasri Department of Computer Science Engineering University of Texas at Arlington, Shamkant B. Navathe College of Computing Georgia Institute of Technology.

https://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam/Database_Systems.pdf

Unit – 2 : Relational Model (12Hrs)

Introduction to relational model, Codd's rules, concepts of domain, attribute, tuple, relation, constraints (Domain, Key constraints, integrity constraints) and their importance , concept of keys (super key, candidate key, primary key, surrogate key, foreign key) , relational Algebra & relational calculus.

Normalization: Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency(1NF, 2NF and 3 NF), Boyce-codd normal form(BCNF)

- **Description:**

This unit focuses on principles and techniques essential for structuring relational databases. It begins with understanding functional dependencies and progresses through normalization processes from First Normal Form (1NF) to Fifth Normal Form (5NF) and Boyce-Codd Normal Form (BCNF). The unit also covers

advanced topics such as handling multivalued and join dependencies to optimize database design for data integrity and query efficiency in various application domains.

- **Learning Outcomes:**

Demonstrate proficiency in identifying functional dependencies, applying normalization techniques (1NF to 5NF), and understanding algorithms for handling multivalued and join dependencies in relational database design

- **Exercises:**
- The marketing company wishes to computerize their operations by using following tables:

CLIENT_MASTER (Client_No, Name, Address1, Address2, City, State, Pincode, Bal_Due)
PRODUCT_MASTER (Product_No, Description, Profit_Percent, Unit_Measure, Qty_On_Hand, Reorder_Lvl, Sell_Price, Cost_Price)
SALESMAN_MASTER (Salesman_No, Name, Address1, Address2, City, State, Pincode, Sal_Amt, Target_Amt, Remarks)
SALES_ORDER(S_Order_No, S_Order_Date, Client_No, Delve_Address, Salesman_No,Delve_Type, Billed_Yn, Delve_Date, Order_Status) SALES_ORDER_DETAILS (S_Order_No, Product_No, Qty_Ordered, Qty_Disp, Product_Rate)
CHALLAN_MASTER (Challan_No, S_Order_No, Challan_Date, Billed_Yn) CHALLAN_DETAILS (Challan_No, Product_No, Qty_Disp)

- **Learning Outcomes:**
- Proficient in using the Entity-Relationship (ER) model for high-level conceptual data modeling and implementing the Relational Data Model to design schemas, enforce constraints, manage updates, transactions, and handle constraint violations effectively in databases.
- **Specific Resources:**
- Data Models, geeksforgeeks, https://www.geeksforgeeks.org/data-models-in-dbms/
- "Understanding SQL and Relational Databases" by Cristian Darie, Karli Watson, Chris Hart, Kevin Hoffman & Julian Skinner.

https://link.springer.com/chapter/10.1007/978-1-4302-0800-6_1

Unit – 3 : Entity Relationship Model: (12Hrs)

Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams

BASIC SQL: Database schema, data types, DDL operations (create, alter, drop, rename), DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, aggregation, grouping, ordering.

- **Description:**

This unit covers Entity-Relationship (ER) modeling for database design, emphasizing entity types, attributes, keys, and weak entity types. It also explores the Relational Data Model, including schema concepts, constraints, update operations, transactions, and managing constraint violations.

- **Learning Outcomes:**
- Proficient in using the Entity-Relationship (ER) model for high-level conceptual data modeling and implementing the Relational Data Model to design schemas, enforce constraints, manage updates, transactions, and handle constraint violations effectively in databases.

Exercises:

- An enterprise wishes to maintain a database to automate its operations. Enterprise divided into two certain departments and each department consists of employees. The following two tables describes the automation schemas:

DEPT (Deptno, Dname, Loc)
EMP (Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, Deptno)

- Design an ER diagram for a hospital management system that includes entities like patients, doctors, and appointments, with appropriate attributes and relationships.
- Identify and define entity types and attributes for a university database system that manages students, courses, and enrolment
- **Specific Resources:**

- "Understanding SQL and Relational Databases" by Cristian Darie, Karli Watson, Chris Hart, Kevin Hoffman & Julian Skinner.
https://link.springer.com/chapter/10.1007/978-1-4302-0800-6_1

Unit – 4: SQL (12Hrs)

Nested queries/ sub queries, implementation of different types of joins, SQL functions(Date, Numeric, String, Conversion functions), Creating tables with relationship, implementation of key and integrity constraints, views, relational set operations , Transaction Control Language: commit, Rollback, Savepoint , DCL :Grant, Revoke

- **Description:**

This unit delves into advanced topics in relational database theory, focusing on the fundamental operations of relational algebra and calculus, including unary and binary relational operations. It also covers SQL standards, providing an in-depth understanding of schema definition, constraints, queries, and views, as well as data manipulation through INSERT, DELETE, and UPDATE statements. The purpose is to equip learners with the knowledge to design, query, and manage relational databases effectively.

- **Learning Outcomes:**

Apply advanced relational algebra and calculus operations to database queries and design, implement, and manage complex schemas, constraints, and data manipulations using SQL-99 standards.

Exercises:

- List the products which have highest sales.
- Find out the details of top 5 earners of company.
- Determine the names of employee, who earn more than their managers.
- Find the names of clients who have placed orders worth of Rs. 10,000/- or more.
- Determine the names of employees, who take highest salary in their departments.
- Find the names of clients who have placed orders before the month of may 2006.
- Find the customer names and address for the clients, who placed the order '019001'.
- Display names of the managers who is having maximum number of employees working under him.
- Create a view, which contain employee names and their manager names working in sales department.
- Find out if the product is '1.44 drive' is ordered by any client and print the client number, name to whom it is sold.
- Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN)
- Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS

  Ex: Select roll number, name of the student who secured fourth rank in the class.
- Queries (along with sub Queries) using UNION, INTERSECT
- (i) Create a user (ii) Create an varray, which holds the employee phone numbers
- (i) Grant Privileges on Tables (ii) Revoke Privileges from Tables

- **Learning Outcomes:**

Apply advanced relational algebra and calculus operations to database queries and design, implement, and manage complex schemas, constraints, and data manipulations using SQL-99 standards.

- **Specific Resources:**
- G. Ozsoyoglu, Z. M. Ozsoyoglu and V. Matos, "Extending relational algebra and relational calculus with set-valued attributes and aggregate functions", https://dl.acm.org/doi/10.1145/32204.32219
- Jan L. Harrington, "SQL Clearly Explained".

https://www.sciencedirect.com/science/article/abs/pii/B9780123756978500017?via%3Dihub

Unit – 5 : PL/SQL (12Hrs)

Introduction , Structure , Control Structures , Cursors , Procedure , Function , Packages , Exception Handling ,Triggers.

- **Description:**

This unit helps to understand the basics of programming and database management, laying the groundwork for more complex concepts.

- **Learning Outcomes:**

These concepts form the basis of PL/SQL programming, allowing you to write efficient and effective database applications

- **Exercises:**
- (i) Lock table in share mode (ii) Lock table in Exclusive mode
- Create a trigger to insert information about the transaction of a customer table. The customer table consists of custno, custname, and money. The information table consists of message field.
- Design a banking application that handles transactions (e.g., deposits, withdrawals) using the Two-Phase Commit Protocol to ensure data consistency across distributed databases.
- Implement a simple online shopping cart system where transactions (e.g., adding items, updating quantities, checkout) are managed, ensuring atomicity and isolation properties.
- **Specific Resources:**
- Dardina Tasmere, Senior Lecturer, Department of Computer Science and Engineering, Bangladesh Army University of Engineering & Technology, Natore, Bangladesh, Md. NazmusSalehinB.Sc Student, Department of Computer Science and Engineering, Bangladesh Army University of Engineering & Technology, Natore, Bangladesh, "Concurrency Control in Database Systems" https://www.cribfb.com/journal/index.php/BJMSR/article/view/365
- SeppoSippu , Eljas Soisalon-Soininen,"Transaction Processing" Management of the Logical Database and its Underlying Physical Structure. https://link.springer.com/book/10.1007/978-3-319-12292-2

Text Books:

1. Raghurama Krishnan, Johannes Gehrke, 2003, Database Management Systems, 3rd Edition, TMH
2. Silberschatz, Korth, 2005, Database System Concepts, 5th edition, TMH

References:

[1] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, (2006), *Database System Concepts.* (6th Ed.) McGraw hill.

[2] Peter Rob, A. Anand Rao, Carlos Coronel, *Database Management Systems*. Cengage Learning

[3] Raghu Ramakrishnan, (2015), *Database Management Systems.* (4th Ed) McGraw-Hill.

[4] Peter Rob & Carlos Coronel, (2008), *Database System Concepts.* Cengage Learning.

[5] **Web Resources:**

[6] **[1]** Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 2013,**"Database System Concepts"**, (6th Edition), McGraw hill,

[7] https://www.amazon.in/Database-System-Concepts-Abraham-Silberschatz/dp/9332901384

[8] **[2]** Elmasri and Navathe : Fundamentals of Database Systems,

[9] https://edurev.in/p/97587/Fundamentals-of-Database-Systems-by-Elmasri--Navat,

[10] https://www.amazon.in/Fundamentals-Database-Systems-Elmasri-Shamkant/dp/B076K8CM55

[11] **[3]** P .S. Gill, Database Management System,

[12] https://www.amazon.in/Database-Management-Systems-P-Gill-ebook/dp/B01GUZBN9K, https://books.google.co.in/books?id=mK4COraJvlIC&printsec=copyright&redir_esc=y#v=onepage&q&f=false

[13] **[4]** Raghu Ramakrishan, Database Management System,

[14] https://www.amazon.in/Database-Management-Systems-Raghu-Ramakrishnan/dp/0072465638,

[15] https://xuanhien.wordpress.com/wp-content/uploads/2011/04/database-management-systems-raghu-ramakrishnan.pdf

***

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:	**23BCMAL232 Database Management Systems**
Offered to:	**BCA Honours**
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 1. A)What are the differences between data and information.(K1)

(OR)

B) Write a short note on evolution of data models. (k2)

Q2 A).Explain different types of attributes with neat diagrams. (k2)

(OR)

B). Explain about different keys in dbms? (k2)

Q3 3. A) Explain about Integrity rules (k2)

(OR)

B) Write about CODD'S rules? (k2)

Q4 4. A) Explain different types of Aggregate functions in SQL. (k2)

(OR)

B) Write a short note on string functions in SQL. (k2)

Q5 5. A) Explain Structure of PL/SQL (k2)

(OR)

B) Explain Functions in PL/SQL (k2)

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

Q6 6. A) Explain the role and advantages of DBMS? (k2)

(OR)

B) Explain briefly about degrees of data abstraction? (k2)

Q7 A) Explain Specialization hierarchy with an example? (k2)

(OR)

B). Explain Entity Relationship diagram with an example (k2)

Q8 A) Write a short note on relational set operators. (k2)

(OR)

B) What is normalization? Explain with an example upto 3NF? (k1)

Q9 A) Explain DDL, DML, DCL commands in SQL with example (k2)

(OR)

B). Explain views in SQL with syntax and examples. (k2)

Q10 A) Discuss about iterative control statements available in PL/SQL with syntax and examples.(k3)

(OR)

B). Explain types of Triggers in PL/SQL (k3)

@@@@@@@

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

Course Code				**23BCMAP232**			
Title of the Course				**Database Management Systems Lab**			
Offered to: (Programme/s)				**BCA Honours**			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**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:
The objective of course is to provide students with practical experience in database management using Oracle SQL and PL/SQL. Students will learn to create and manage database objects, perform data manipulation and retrieval, implement advanced querying techniques, and develop PL/SQL programs

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Introduce students to the foundational concepts and syntax of SQL
2	Equip students with the skills to design and manage relational databases
3	Develop students' ability to perform complex data retrieval and manipulation.
4	Provide comprehensive training in procedural programming using PL/SQL
5	Teach students how to manage errors and optimize database performance

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Using DDL commands in Oracle, including creating, altering, and dropping tables	**K2**	**6,7**	**1,2**
CO2	Performing data manipulation operations using DML commands	**K3**	**6,7**	**1,2**
CO3	Understand and implement various types of joins	**K3**	**6,7**	**1,2**
CO4	Write and execute basic PL/SQL programs	**K3**	**6,7**	**1,2**
CO5	Use both implicit and explicit cursors in Oracle PL/SQL, execute triggers	**K3**	**6,7**	**1,2**

CO-PO MATRIX									
CO NO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PSO1**	**PSO2**
CO1						**2**	**3**	**2**	**2**
CO2						**3**	**2**	**2**	**3**
CO3					**3**	**2**	**3**	**3**	**2**
CO4						**3**	**3**	**2**	**3**
CO5						**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 Database management systems lab course, providing hands-on practice with Oracle technology

Unit 1: Implementing DDL commands in Oracle (6Hrs)

Lab 1:

Exercise 1: Creating Tables without Constraints

1. Create tables without applying any constraints to understand basic table creation.

Tasks:

- Create a table Employees with columns: Employee_ID, First_Name, Last_Name, Hire_Date, and Department.
- Create a table Projects with columns: Project_ID, Project_Name, and Start_Date.

2. Creating Tables with Primary Key and Foreign Key Constraints

Exercise 2:

Defining Tables with Primary and Foreign Keys

1. **Objective:** Learn to create tables with primary key and foreign key constraints to ensure referential integrity.
2. **Tasks:**

1. Create a table Departments with columns: Department_ID and Department_Name, and apply a primary key constraint on Department_ID.

2. Create a table Employees with columns: Employee_ID, First_Name, Last_Name, Hire_Date, Department_ID, and apply a primary key constraint on Employee_ID. Add a foreign key constraint on Department_ID to reference Departments.

Lab 2:

Exercise 3: Creating Tables with Unique and Check Constraints

Objective: Create tables with unique and check constraints to enforce data uniqueness and valid data values.

1. **Tasks:**

1. Create a table Products with columns: Product_ID, Product_Name, Price, and Category. Apply a primary key constraint on Product_ID and a unique constraint on Product_Name.

2. Create a table Orders with columns: Order_ID, Order_Date, Product_ID, and Quantity. Apply a primary key constraint on Order_ID and a check constraint to ensure Quantity is greater than 0.

Exercise 4: Creating Tables with Composite Keys and Default Values

1. **Objective:** Create tables with composite primary keys and default values for columns.
2. **Tasks:**

- Create a table Order_Items with columns: Order_ID, Product_ID, Quantity, and Price. Apply a composite primary key constraint on Order_ID and Product_ID. Set default values for Quantity (1) and Price (0.00).
- Create a table Customer_Reviews with columns: Review_ID, Customer_ID, Review_Date, and Rating. Apply a primary key constraint on Review_ID and set a default value for Review_Date as the current date.

Exercise 5: Creating Tables with Referential Integrity Constraints

1. **Objective:** Create tables that enforce referential integrity between parent and child tables.
2. **Tasks:**

- Create a table Customers with columns: Customer_ID, Customer_Name, and Contact_Number, and apply a primary key constraint on Customer_ID.
- Create a table Invoices with columns: Invoice_ID, Customer_ID, Invoice_Date, and Amount. Apply a primary key constraint on Invoice_ID and a foreign key constraint on Customer_ID to reference Customers.

Unit 2: DML commands

Lab 3:
Exercise 6:

- Insert Data into emp and dept tables

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

	A	B	C	D	E	F	G	H
1	7839	KING	PRESIDENT		1981-11-1	5000		10
2	7698	BLAKE	MANAGEF	7839	1981-05-0	2850		30
3	7782	CLARK	MANAGEF	7839	1981-06-0	2450		10
4	7566	JONES	MANAGEF	7839	1981-04-0	2975		20
5	7788	SCOTT	ANALYST	7566	1982-12-0	3000		20
6	7902	FORD	ANALYST	7566	1981-12-0	3000		20
7	7369	SMITH	CLERK	7902	1980-12-1	800		20
8	7499	ALLEN	SALESMAI	7698	1981-02-2	1600	300	30
9	7521	WARD	SALESMAI	7698	1981-02-2	1250	500	30
10	7654	MARTIN	SALESMAI	7698	1981-09-2	1250	1400	30
11	7844	TURNER	SALESMAI	7698	1981-09-0	1500	0	30
12	7876	ADAMS	CLERK	7788	1983-01-1	1100		20
13	7900	JAMES	CLERK	7698	1981-12-0	950		30
14	7934	MILLER	CLERK	7782	1982-01-2	1300		10

https://livesql.oracle.com/apex/livesql/file/content_O5AEB2HE08PYEPTGCFLZU9YCV.html

Lab 4: Queries
Exercise 7:

1. Display all the information of the EMP table?
2. Display unique Jobs from EMP table?
3. List the emps in the asc order of their Salaries?
4. List the details of the emps in asc order of the Dptnos and desc of Jobs?
5. Display all the unique job groups in the descending order?
6. Display all the details of all 'Mgrs'
7. List the emps who joined before 1981.
8. List the Empno, Ename, Sal, Daily sal of all emps in the asc order of Annsal
9. Display the Empno, Ename, job, Hiredate, Exp of all Mgrs
10. List the Empno, Ename, Sal, Exp of all emps working for Mgr 7369. 11. Display all the details of the emps whose Comm. Is more than their Sal.
12. List the emps in the asc order of Designations of those joined after the second half of 1981.
13. List the emps along with their Exp and Daily Sal is more than Rs.100.
14. List the emps who are either 'CLERK' or 'ANALYST' in the Desc order.

Unit 3: Joins and views
Lab 5: joins
Exercise 8:

15. List the total information of EMP table along with DNAME and Loc of all the emps Working Under 'ACCOUNTING' & 'RESEARCH' in the asc Deptno.

16. List the Empno, Ename, Sal, Dname of all the 'MGRS' and 'ANALYST' working in New York, Dallas with an exp more than 7 years without receiving the Comm asc order of Loc.
17. Display the Empno, Ename, Sal, Dname, Loc, Deptno, Job of all emps working at CJICAGO or working for ACCOUNTING dept with Ann Sal>28000, but the Sal should not be=3000 or 2800 who doesn't belongs to the Mgr and whose no is having a digit '7' or '8' in 3rd position in the asc order of Deptno and desc order of job.
18. Display the total information of the emps along with Grades in the asc order.
19. List all the Grade2 and Grade 3 emps
20. Display all Grade 4,5 Analyst and Mgr.
21. List the Empno, Ename, Sal, Dname, Grade, Exp, and Ann Sal of emps working for Dept10 or20.

Lab 6: views

Exercise 9

1. Create a simple view to display specific columns from a table.

Task: Create a view named Employee_View that displays Employee_ID, First_Name, and Last_Name from the Employees table.

2. Create a view that joins multiple tables.

Task: Create a view named Employee_Department_View that displays Employee_ID, First_Name, Last_Name, and Department_Name by joining the Employees and Departments tables.

3. Create an updatable view that allows DML operations.

Task: Create a view named Updateable_Employee_View that displays Employee_ID, First_Name, and Last_Name and allows updates to the Last_Name column.

4. Create a view that is read-only and does not allow DML operations.

Task: Create a view named ReadOnly_Department_View that displays Department_ID and Department_Name.

Unit 4: joins, subqueries, nested subqueries

Lab 7:

Exercise 10:

1. List the total information of EMP table along with DNAME and Loc of all the emps Working Under 'ACCOUNTING' & 'RESEARCH' in the asc order of Deptno.

2. List the Empno, Ename, Sal, Dname of all the 'MGRS' and 'ANALYST' working in New York, Dallas with an experience more than 7 years without receiving the Comm asc order of Loc.

3. List the Empno, Ename, Sal, Dname, Grade, Exp, and Ann Sal of emps working for Dept10 or20.

4. List the details of the Depts along with Empno, Ename or without the emps

5. List the details of the emps whose Salaries more than the employee BLAKE

6. List the Emps who are senior to their own MGRS

7. List the emps whose jobs same as SMITH or ALLEN

Lab 8:

Exercise 11:

1. **Create some sub-quries on emp and dept tables.**
2. **Create some nested subq-ueries on emp, dept tables.**
3. Concatenation of strings
4. Finding the length of a string
5. Substring extraction
6. String replacement
7. String case conversion

Unit 5: PL/SQl programs, triggers and exception handling (6Hrs)

Lab 9: basic PL/SQL programs

Exercise 12:

1. Write programs on Nested Blocks and Control Structures

2. Display Employee Details Using PL/SQL

3. Write a program to check the given number is prime or not.

Lab 10: Triggers and exceptions

Exercise 13:

1. Create a Trigger to Automatically Update Book Quantity When a Book is Returned
2. Create a Trigger to Update `Last_Updated` Column
3. Test the Trigger
4. Handle Exception for Division by Zero
5. Handle Exception for No Data Found
6. Create a user defined exception

References:

2. Nilesh Shah. (2011). *Database Systems Using ORACLE* (2nd ed.). PHI
3. https://www.youtube.com/playlist?list=PLL_LQvNX4xKyiExzq9GKwORoH6nvaRnOQ

***

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

23BCMAP232 : **Database Management Systems Lab**

Offered to:BCA.Hons **Semester: III**
Max. Marks : 50 (CIA: 15 + SEE: 35) **Hrs/Week: 2**

Model Paper : Practicals

Time: 3 Hrs. **Max. Marks: 35**

Section – A

1. Experiment-1 15 M
2. Experiment-2 10 M

Section – B

Viva Voce 10 M

@@@@

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

Course Code				23BCMAL233			
Title of the Course				**Object Oriented Programming Using Java**			
Offered to: (Programme/s)				**BCA Honours**			
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:				**MAJOR**			
Crosscutting Issues of the Course:							
Pre-requisites, if any				Knowledge in C Programming			

Course Description:

This course provides the fundamental components and libraries of the Java programming language, with a strong emphasis on object-oriented programming (OOP) principles. It constitutes as the foundation for Java development, providing the essential building blocks and features for creating robust and scalable applications.

Course Aims & Objectives:

S. No	COURSE OBJECTIVES
1	Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
2	Realize fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
3	Analyze step by step and develop programs on inheritance and interfaces, arrays and string handling functions
4	Understand the Fundamental features of multi-threaded programs, Exception handling and packages.
5	Understand the principles of applets, I/O streams in java and java database connectivity

Course Outcomes:

At the end of the course, the student will / will be…

NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Develop a comprehensive understanding how object-oriented concepts are incorporated into the Java programming language	K2	**1,2,6**	**1,2**
CO2	Implementing Object Oriented Programming Concepts(class, constructor, overloading) in java	K3	**2,6**	**1,2**
CO3	Implementing inheritance and interfaces in a Java program.	K3	**2,6**	**1,2**
CO4	Implementing Multithreading, exception handling and packages in Java.	K3	**2,6**	**1,2**
CO5	Implementing Applets, Files and Jdbc Connectivity in Java programs.	K3	**2,6**	**1,2**

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	**3**	**3**				**2**		**2**	**1**
CO2		**3**				**2**		**2**	**1**
CO3		**3**				**2**		**2**	**1**
CO4		**3**				**2**		**2**	**1**

CO5		**3**				**2**		**2**	**1**

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.Debasis Samanta, 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: Arrays, Strings, Inheritance 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:
3. Types of inheritances.
4. String handling functions and array creation in java

Exercises:
1. **Design a Java program to calculate multiplication of 2 matrices**
2. **Develop a program on Multiple Inheritance.**

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.Debasis Samanta, 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. **Develop a program to create and Import Packages**
2. **Construct Java programs to implement various types of Exception Handling Mechanisms**

Learning Outcomes:

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

Resources:

1.Prof.Debasis Samanta, 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.Debasis Samanta, 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.Debasis Samanta, 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: Applets, Streams, I/O Files and Jdbc (12 Hrs.)

Local and remote applets-Applets and Applications-Building Applet code- Applet Life cycle: Initialization state, Running state, Idle or stopped state, Dead state, Display state-Concept of Streams-Stream classes-Byte Stream Classes-Character Stream classes: Reader stream classes, Writer Stream classes-Reading and writing files.

Jdbc introduction-stages in Jdbc program-working with oracle database: inserting, updating and deleting records.

Description:
This unit focuses Understanding the principles of applets, I/O streams in java and java database connectivity
Examples:
1. Applet creation
2. Writing and Reading Files.
3. JDBC Connectivity

Exercises:
1. **Design a program to create an Applet**
2. Create a program for writing and reading Files.

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:

1.Prof.Debasis Samanta, Dept of Computer science, IIT Kharagpur."Applet Programming in Java", 2018. https://www.youtube.com/watch?v=cC_Ij7WmP_k&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=34

2.Prof.Debasis Samanta, Dept of Computer science, IIT Kharagpur." JDBC", 2018. https://www.youtube.com/watch?v=ajhWv31oN1k&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho&index=50

3. "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:
6. Programming in Java by Sachin Malhotra, OXFORD University Press
7. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TATA McGraw-Hill Company.

8. Deitel &Deitel. Java TM: How to Program, PHI (2007)
9. Java Programming: From Problem Analysis to Program Design- D.S Mallik
10. Object Oriented Programming Through Java by P. Radha Krishna, Universities Press (2008)

Web Resources:

Prof.Debasis Samanta, 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 STRUCTURE

Course Code & Title of the Course:	**23BCMAL233** **Object Oriented Programming Using Java**
Offered to:	**BCA Honours**
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.

1. a)Discuss about JVM. k2

   OR

   b)Explain command line arguments with an example k2
2. a) Explain method overloading with an example. k2

   OR

   b)Discuss concept of static members in java with example. k2
3. a)Discuss about final class k2

   b)Illustrate implementing interfaces in java with example. k3
4. a) Describe creating threads in java with an example k4

   OR

   b)Describe package creation and accessing with example.k4
5. a) Explain character stream classes in java. k2

   OR

b) Explain applet creation with example. k2

Section B: Long Answer Questions (50 Marks)

Answer All questions. Each question carries 10 Marks.

6. a) Discuss Object Oriented Programming Principles. k2)

   (OR)

   b)Discuss Java Buzz words.k2)
7. a) Explain accessing class members with an example. k2

   (OR)

   b)Explain Constructor with an example. k2
8. a) Illustrate string handling methods in java with examples. k3

   (OR)

   b)List of different types of inheritance in java and explain with examples. k2
9. a) Explain life cycle of a thread with neat diagram. k2

   (OR)

b) Define Exception. Explain Exception handling mechanism in java with examples k2

10. a) Explain life cycle of applet with neat diagram. k2

    10. (OR)

    b)Explain different stages in JDBC program with an example. (CO5,L2)

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				**23BCMAP233**			
Title of the Course				**Object Oriented Programming in Java Lab**			
Offered to: (Programme/s)				**BCA Honours**			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:		**2024-25**		**Semester:**			**3**
Course Category:		**Major**		**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:
The objective of course is to provide students with practical experience in Object Oriented Programming in Java.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Teach students to know the fundamental concepts in java
2	Provide comprehensive training in designing classes, objects and methods in java
3	Teach students to know inheritance, interfaces concepts in java
4	Train students to gain knowledge in multi threading , exception handling and packages
5	Train students to know Applets Creation,File Creation and JDBC Connectivity

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Creating java programs that covers fundamental concepts	**K6**	**1,2,6**	**1,2**
CO2	Creating class, constructor, method overloading, method overriding in java.	**K6**	**2,6**	**1,2**
CO3	Creating different types of inheritance and interfaces in a Java program	**K6**	**2,6**	**1,2**
CO4	Creating Multithreading, different types of exception handling mechanisms, Creating and accessing packages in Java.	**K6**	**2,6**	**1,2**
CO5	Creating Applets,files and JDBC Connectivity in Java program.	**K6**	**2,6**	**1,2**

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

Lab 1

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.

Unit-2: Control statements, Classes, Objects and Methods

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.**

4. **Construct a Java program that implements method overloading.**

Unit-3: Inheritance, Arrays, Strings and Interfaces

Lab 3

5. **Design a Java program to calculate multiplication of 2 matrices.**

6. **Construct Java programs to implement various types of inheritance**

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

Lab 4

7. **Write a java program to implement runtime polymorphism.**

8. **Develop java program to implement Abstract Classes and Final Keyword**

Lab 5

9. **Design a program for implementing interfaces.**

10. **Develop a program on Multiple Inheritance.**

Unit-4: Multi-Threading, Exception Handling and Packages

Lab 6

11. **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.**

12. **Develop 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.**

Lab 7

13. **Develop 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.**

14. **Construct program to create and Import Packages**

Lab 8

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

iv. **Arithmetic Exception**

v. **Number Format Exception**

vi. **Array Index Out of Bounds Exception**

16. Design a program to demonstrate Catch Blocks

Unit-5: Applets, Streams, Files and Jdbc

Lab -9

17. **Design a program to create an Applet**

18. Create a program for writing and reading Files.

Lab -10

19. **Design a program to insert records in DB table using JDBC.**

20. **Develop a program to Retrieve records from DB table using JDBC**

References:

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

Web Resources:

Prof.Debasis Samanta, 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)

23BCMAP233: Object Oriented Programming Using Java

Offered to: BCA Honours **Semester: III**
Max. Marks: 50 (CIA: 15 + SEE: 35) **Hrs/Week: 2**

Model Paper: Practicals

Time: 3 Hrs. **Max. Marks: 35**

Section – A

1. Experiment-1 15 M
2. Experiment-2 10 M

Section – B

Viva Voce 10 M

@@@@

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

Course Code				**23BCMAL234**			
Title of the Course				**SOFTWARE ENGINEERING**			
Offered to: (Programme/s)				**B.C.A Honours**			
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:				Employability/ Skill development			
Crosscutting Issues of the Course :							
Pre-requisites, if any							

Legend:
Offered to: B.C.A Hons
Category: Major
Course Relates to: Local, Regional, National, Global
Type of the Course: Employability/ Skill development
Crosscutting Issues of the Course: Gender, Environment and Sustainability, Human Values and Professional Ethics
L: Lecture; T: Tutorial; P: Practicum/Practical/Project; C: Credits

Course Description:
The course is to assist the student in understanding the basic theory of software engineering, and to apply these basic theoretical principles to a group software development project.

Program Design Tools:
1. To draw dataflow diagrams using Microsoft Visio Software, SmartDraw,
2. To draw UML diagrams using Rational Rose Software, Star UML, etc.

Course Aims and Objectives:

S. NO	COURSE OBJECTIVES
1	Grasp fundamental software engineering concepts, methodologies, and principles
2	Known about ethical responsibilities of software engineers.
3	Gain the ability to design software systems that are modular, scalable, and maintainable.
4	Study the cognitive, physical, and social aspects of human interaction with technology.
5	Learn techniques for software testing and quality assurance and theoretical knowledge to real-world scenarios through case studies and practical exercises..

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Understand the requirements of the software projects.	K2	PO5,PO7	PSO2
CO2	Ability to analyze software requirements with existing tools	K4	PO5,PO7	PSO1PSO,2
CO3	Apply different testing methodologies	K3	PO5,PO7	PSO1,PSO2

CO4	Understand and apply the basic project management practices in real life projects	K2,K4	PO5,PO7	PSO1,PSO2
CO5	Apply on software projects	K4	PO5,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
CO2					3		2	3	2
CO3					3		2	2	
CO4					3		2	2	2
CO5					3		2	3	2

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

Course Structure:

Unit – 1 : (10Hrs)

Introduction to Software Engineering: The Software Engineering – Evolution and impact, Software Development Projects, Software Process and Project Metrics, Emergence of Software Engineering, Computer Systems Engineering,

Software Life cycle models: Need for life Cycle model, classical waterfall model, Iterative waterfall model, V-model, Prototyping model, Evolutionary model, Spiral model, Agile Development Models, Comparison of different life cycle models.

Description:

Software Engineering is the systematic application of engineering principles to the development, operation, maintenance, and retirement of software. It's a discipline focused on producing high-quality software that meets user needs within budget and on time.

Learning Outcomes:

- Understanding software project management principles.
- Proficiency in software development tools and environments.
- Knowledge of software design patterns and architectures.

Examples:

Conduct workshops and training sessions to ensure all team members understand the principles and practices of software engineering. Create a repository of reference materials and best practices

Exercises

Analyze the size factors of a software project Building online stores, payment gateways

Specific Resources: (web)

Prof. Rajib Mall, Assistant Professor, Department of Computer Science and Engineering, IIT Kharagpur

https://youtu.be/Ln_LP7c23WM

UNIT – II (12Hrs)

Software Project Management: Responsibilities of a Software Project Manager, Project planning, Metrics for Project size estimation and scheduling.

Requirement Analysis: Requirements gathering and analysis, Software Requirements Specification ___ contents of the SRS document, Functional requirements, Traceability, Characteristics of good SRS DOCUMENT, Organization of the SRS document.

Description:

Software cost estimation is the process of predicting the financial resources required to develop and maintain a software system. Accurate cost estimation is crucial for project planning, budgeting, and resource allocation. Various techniques are employed to estimate software costs

Learning Outcomes:

- Understand the concept of software cost estimation

- Identify key factors
- Apply various cost estimation techniques
- Estimate staffing levels
- Determine the factors influencing software maintenance costs
- Create comprehensive software requirements specifications
- Utilize formal specification techniques
- Select appropriate languages and processors
- Evaluate the impact of software requirements on project costs

Examples:

- Detailed Explanation of the COCOMO Model

Exercises:

- The basic COCOMO formula is:

$\text{Effort}=a\times(\text{KLOC})^b$

Where:

- **Effort** is the number of person-months.
- **KLOC** is the estimated number of thousands of lines of code.
- **a** and **b** are coefficients that vary depending on the project type (organic, semi-detached, or embedded).

For example, for an organic project: $a=2.4, b=1.05a = 2.4, \backslash, b = 1.05a=2.4,b=1.05$

Suppose we estimate the size of the software to be 50 KLOC:

Specific Resources: (web)

Prof. Rajib Mall, Assistant Professor, Department of Computer Science and Engineering, IIT Kharagpur

https://youtu.be/ilHPCbkZLV4

UNIT – III (12Hrs)

Software design: Desirable characteristics of a good software design, Cohesion and coupling, Layer Arrangement of Modules, Function-oriented design and Object-oriented design.

Function-oriented software Design: Overview of SA/SD methodology, structured analysis, Data Flow Diagrams, Structured Design and Detailed Design.

Description:

Software design is the process of conceptualizing the software solution to a problem, transforming user requirements into a suitable form, and producing a design document based on the customer requirements. It's a crucial phase that bridges the gap between requirements analysis and software implementation.

Learning Outcomes:

- Define software design
- Transform user requirements
- Apply fundamental design concepts
- Utilize various design notations
- Employ different design techniques
- Conduct detailed design
- Design software for real-time and distributed systems
- Develop comprehensive test plans to ensure software quality.
- Participate in software design reviews

Examples:

- Software Design for an E-Commerce Platform

Exercises:

- Represent Design Notations for e-commeerce platform

UML Diagrams

- **Class Diagrams**: Represent classes and relationships in each module.
- **Sequence Diagrams**: Show interactions between objects during user registration, product search, order placement, and payment processing.
- **Data Flow Diagrams (DFDs)**

- Illustrate how data flows through the system, from user inputs to database storage and retrieval.

Specific Resources: (web)

Prof. Mythii Vutukuru, assistant Professor, IIT Bombay, software Design, https://youtu.be/3fLahzQr8EI?list=PLDW872573QAZNlUzWVzoU8cCadXg1NUGK

UNIT IV (12Hrs)

User interface design: Characteristics of good user interface design, Basic concepts, Types of user interfaces, component-based GUI development, A user interface Design Methodology

Unified Modeling Language: Overview of Object-oriented concepts, Unified Modeling Language, UML diagrams, use case model class diagrams, Interaction diagrams, Activity diagrams, state chart diagrams

Description:

User Interface (UI) design is the process of creating effective interactions between humans and computer systems. It's about designing the look, feel, and behavior of software applications to ensure they are user-friendly, efficient, and enjoyable to use.

Learning Outcomes:

- Define user interface design.
- Apply human factors principles.
- Understand the fundamentals of human-computer interaction.
- Develop effective user interfaces.
- Create visually appealing and intuitive user interfaces.
- Evaluate user interface designs.
- Design user interfaces for specific user groups.
- Apply user interface design principles to real-time systems .
- Stay updated on emerging trends and technologies.

Examples:

- User Interface Design for a Real-Time Patient Monitoring System

Exercises:

Example Interface Design:

Dashboard:

- **Patient List**: A list of all monitored patients with summary information (name, room number, key vital signs).
- **Critical Alerts**: A section for critical alerts, sorted by severity and time.
- **Navigation**: Easy access to patient detail views, settings, and system logs.

Patient Detail View:

- **Vital Signs Graphs**: Real-time graphs showing trends for heart rate, blood pressure, temperature, etc.
- **Alerts History**: A log of all alerts for the patient, with timestamps and statuses.
- **Actions**: Buttons for common actions, such as acknowledging alerts, adding notes, or calling for assistance.

Settings and Customization:

- **Alert Thresholds**: Interface for setting and adjusting alert thresholds for different vital signs.
- **Display Options**: Options for customizing the layout, themes, and data visibility.

Specific Resources: (web)

Dr. Samit Bhattacharya, Assistant Professor, Computer Science and Engineering, IIT GUWAHATI, Design & Implementation of Human-Computer Interfaces

https://youtu.be/uFYuHHglC6U?list=PLwdnzlV3ogoVKbbd4bwgSoga7EEuX5kFf

UNIT V (14Hrs)

Software quality and testing: Software Quality Assurance - Quality metrics - Software Reliability - Software testing - Path testing – Control Structures testing - Black Box testing - Integration, Validation and system testing - Reverse Engineering and Reengineering.

CASE Tools: Projects management, tools - analysis and design tools – programming tools - integration and testing tool - Case studies.

Description:

software quality and testing are critical for producing reliable and efficient software. CASE tools can significantly enhance the software development process by automating tasks and improving productivity

Learning Outcomes:

- Define software quality assurance.
- Identify and apply quality metric.
- Explain the concept of software reliabilit.
- Design and execute various software testing techniques.
- Conduct integration, validation, and system testing.
- Apply reverse engineering and reengineering techniques to analyze and modify existing software systems.
- Utilize CASE tools to support software quality assurance and testing activities.
- Create and manage software test plans and test cases.
- Analyze test results and generate test reports to identify defects and recommend corrective actions.
- Understand the importance of software quality

Examples:

- Implementing Test Automation for a E-commerce Platform

Exercises:

- **Client:** A large online retailer looking to improve software quality and reduce testing time.
- **Project:** Implement a test automation framework for functional, regression, and performance testing of their e-commerce platform.

Specific Resources: (web)

Prof. Rajib Mall, Assistant Professor, Department of Computer Science and Engineering, IIT Kharagpur

https://youtu.be/ilHPCbkZLV4

Choose any two of above case studies and do the following exercises for that Case Study

1. Write the software requirements specification document
2. Draw the entity relationship diagram
3. Draw the data flow diagrams
4. Draw use case diagrams
5. Draw activity diagrams for all use cases
6. Draw sequence diagrams for all use cases
7. Draw collaboration diagram
8. Assign objects in sequence diagrams to classes and make class diagram.

Student Activity:

1. Visit any financial organization nearby and prepare requirement analysis report
2. Visit any industrial organization and prepare risk chart

Text Books:

1. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI

References:

1. R.Fairley, Software Engineering Concepts, Tata McGraw-Hill, 1997.
2. Software Engineering, H. Sommervill Ian , Addition Wesley Pub. Co.
3. Software Engineering: An object Oriented Perspective by Braude, E.J., Willey, 2001

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:	**23BCMAL234**
Offered to:	**B.C.A Honours**
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) Why Software Engineering is required?(K2)
OR
(b) What are the phases of the waterfall model? (K1)

Q2 (a) Summarize responsibilities of software manager. (K3)
OR
(b) Summarize Characteristics of SRS Document. (K3)

Q3 (a) What is DFD? Explain with an example. (K2)
OR
(b) Write about Modular Design. (K1)

Q4 (a) Differentiate Graphical user interface and text - based user interface. (K3)
OR
(b) List and explain various types of user interfaces. (K2)

Q5 (a) Write about Glass Box testing. (K1)
OR
(b) Summarize Quality metrics (K3)

Section B: Long Answer Questions (50 Marks)

Answer All questions. Each question carries 10 Marks.

Q6 (a) Compare different software development life cycle modes. (K3)
OR
(b) Explain about spiral model. (K2)

Q7 (a) What are the reas6ns to spend time and resources to develop a SRS document. (K1)
OR
(b) Explain metrics for project size estimation. (K2)

Q8 (a) Explain about coupling and cohesion. (K2)
OR
(b) Write about overview of SA/SD methodologies. (K1)

Q9 (a) Explain different types of user interfaces (K2)
OR
(b) Explain IJML class diagrams with an example (K2)

Q10 (a) What is program testing? Briefly explain the following tests:,
a) Unit testing.
b) Block box testing
c) White box testing (K1)
OR
(b) Discuss analysis and design tools. (K2)

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

Course Code				**23BCMAP234**			
Title of the Course				**SOFTWARE ENGINEERING LAB**			
Offered to: (Programme/s)				**BCA Honours**			
L	**0**	**T**	**0**	**P**	**2**	**C**	**1**
Year of Introduction:	**2024-25**			**Semester:**			**3**
Course Category:	**Major**			**Course Relates to:**	**Global**		
Year of Revision:				**Percentage:**			
Type of the Course:				**Employability**			
Crosscutting Issues of the Course:				**L, P, C**			
Pre-requisites, if any							

Course Description:
This course provides basic an opportunity to practically implement various OOSE concepts using various case studies. This course enables students to analyse and design the system in object-oriented manner using Eclipse tool.

Course Aims and Objectives:

S.NO	COURSE OBJECTIVES
1	Understand the basics and planning of a software project
2	Analyse software cost estimation and its techniques
3	Software Design
4	User interface design
5	Software testing and validations

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

CO NO	COURSE OUTCOME	BTL	PO	PSO
CO1	Understand the requirements of the software projects.	**K2**	**5,7**	**2**
CO2	Ability to analyze software requirements with existing tools	**K4**	**5,7**	**1,2**
CO3	Apply different testing methodologies	**K3**	**5,7**	**1,2**
CO4	Understand and apply the basic project management practices in real life projects	**K2,K4**	**5,7**	**1,2**
CO5	Apply on software projects	**K4**	**5,7**	**1,2**

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**
CO2					**3**		**2**	**3**	**2**
CO3					**3**		**2**	**2**	
CO4					**3**		**2**	**2**	**2**
CO5					**3**		**2**	**3**	**2**

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 Software Engineering course, providing hands-on practice with Eclipse UML2/any other Open Source Tools

Design Following Systems in Object Oriented Approach using UML with open source tools (Eclipse UML2 or any other Open source tools):

1. Online Examination System.
2. Online Railway Reservation.
3. Library Maintenance System.
4. Any E-Commerce Portal.
5. Biometric Attendance System.

1. Write down the problem statement for a suggested system of relevance.
2. Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.
3. To perform the function oriented diagram: Data Flow Diagram (DFD) and Structured chart.
4. To perform the user‘s view analysis for the suggested system: Use case diagram.
5. To draw the structural view diagram for the system: Class diagram, object diagram.
6. To draw the behavioral view diagram : State-chart diagram, Activity diagram
7. To perform the behavioral view diagram for the suggested system : Sequence diagram, Collaboration diagram
8. To perform the implementation view diagram: Component diagram for the system.
9. To perform the environmental view diagram: Deployment diagram for the system.
10. To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.
11. Perform Estimation of effort using FP Estimation for chosen system.
12. To Prepare time line chart/Gantt Chart/PERT Chart for selected software project

Note: Student is expected to analyze the system in object oriented manner and design the system in object oriented approach using UML with open source tools

References:

1. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI
2. R.S. Pressman, Software Engineering a practitioner’s approach, Fourth Ed., McGraw Hill, 1997

***

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

23BCMAP234: **Software Engineering Lab**

Offered to: B. Sc. Hons. CSCS **Semester: III**
Max. Marks: 50 (CIA: 15 + SEE: 35) **Hrs/Week: 2**

Model Paper: Practicals

Time: 3 Hrs. **Max. Marks: 35**

Section – A

1. Experiment-1 15 M
2. Experiment-2 10 M

Section – B

Viva Voce 10 M

@@@@