ANNEXURE III

LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHEME OF INSTRUCTIONS & EXAMINATIONS [LR-24]

(W. e. f Academic Year 2025-26)

II-B.E. III-Semester

S. No.	Course Code	Category	Course Title	Scheme of Instructions				Scheme of Examination			CREDITS
				L	T	P/D	Contact Hours/Week	Maximum Marks		Duration in Hours	
								CIE	SEE		
Theory Course											
1	U24ME304	ESC	Operations Research Techniques	3	-	-	3	40	60	3	3
2	U24CD303	PCC	Software Engineering	2	1	-	3	40	60	3	3
3	U24CS301	PCC	Discrete Mathematics	3	-	-	3	40	60	3	3
4	U24CS302	PCC	Data Structures #	3	-	-	3	40	60	3	3
5	U24IT301	PCC	Data Base Management System	3	-	-	3	40	60	3	3
Practical/ Laboratory Course											
6	U24CS3L1	PCC	Data Structures Lab	0	0	3	3	25	50	3	1.5
7	U24IT3L1	PCC	Data Base Management System Lab	0	0	3	3	25	50	3	1.5
8	U24CS3L3	PCC	Data Visualization lab	0	0	3	3	25	50	3	1.5
9	U24EP3L1	HSMC	Design Thinking Lab	-	-	2	2	50	-	-	1
For Lateral Entry Students Only											
Bridge Course*/ Mandatory Courses*											
9	U24CS3L2	ESC	C Programming Lab	-	-	2	2	50	-	2	-
10	U24EN2L1	HSMC	Effective Communication and Soft Skills Lab	-	-	3	3	50	-	3	-
11	U24CH202	MC	Environmental Science	2	-	-	2	40	60	3	-
Total				**13**	**1**	**11 (15*)**	**26 (34*)**	**325 (505*)**	**450 (570*)**	**--**	**20.5**

*** : Bridge Course Only for Lateral Entry admitted Students.**

L: Lecture *(Hrs/Wk/Sem)* **T**: Tutorial *(Hrs/Wk/Sem)* **P**: Practical **D**: Drawing *(Hrs/Wk/Sem)*

 # : Term work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

CIE: Continuous Internal Evaluation **SEE**: Semester End Examination **PCC:** Program Core Course

ME: Mechanical Engineering. **PROJ:** Project **ESC:** Engineering Science Course

CM: CSE-AIML **EC**: Electronics Communication **CS**: Computer Science

EN: English

Course Code	Course Title						Core/Elective
U24ME304	**Operations Research Techniques**						**Core**
Prerequisite	Hours Per Week				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
Basic Maths	**3**	**-**	**-**	**-**	**40**	**60**	**3**

Course Objectives:

Develop ability to

1. Explain with examples, the basic terminology of functions, relations, and sets.
2. Perform the operations associated with sets, functions, and relations.
3. Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
4. Describe the importance and limitations of predicate logic.
5. Use Graph Theory for solving problems.

Course Outcomes:

At the end of the course, student would be able to

1. Understand the ideas of mathematical induction to recursion and recursively defined structures.
2. Prepare the students to have the knowledge of Linear Programming Problem in Operations
3. Research at the end students would be able to understand the concept and develop the models for different applications.
4. Make students understand the concept Replacement models at the end students would able to explain various features and applications of replacement models in real time scenario.
5. Prepare the students to understand theory of Game in operations research at the end students would able to explain application of Game theory in decision making for a conflict

UNIT – I

Introduction: Definition and Scope of Operations Research.

Linear Programming: Introduction, Formulation of linear programming problems, graphical method of solving LP problem, simplex method, maximization and minimization, Degeneracy in LPP, Unbounded and, Infeasible solutions.

UNIT – II

Transportation Models: Finding an initial feasible solution - North West corner method, least cost method, Vogel‘s Approximation method, Finding the optimal solution, optimal solution by stepping stone and MODI methods, Special cases in Transportation problems - Unbalanced Transportation problem.

Assignment Problems: Hungarian method of Assignment problem, Maximization in Assignment Problem, unbalanced problem, problems with restrictions, travelling salesman problems.

UNIT – III

Network Fundamentals- scheduling the activities -Fulkerson’s Rule –CPM- earliest and latest times -determination of ES and EF in the Forward Pass - LS and LF in backward pass determination of Critical Path, Crashing, time cost trade off. PERT-Beta Distribution, probabilistic models, Calculation of CP, resource analysis and allocation.

UNIT – IV

Replacement Models: Introduction, replacement of items that deteriorate ignoring change in money value, replacement of items that deteriorate considering change in money value with time, replacement of items that fail suddenly - Individual replacement policy, Group replacement policy.

Game Theory: Introduction, 2-person zero sum games, Maximin - Minimax principle, Principle of Dominance, Solution for mixed strategy problems, Graphical method for 2 x n and m x 2 games.

UNIT – V

Sequencing Models: Introduction, General assumptions, processing n jobs through 2 machines, processing n jobs through m machines, Processing 2 jobs through m machines

Queuing Theory: Introduction, single channel - Poisson arrivals - exponential service times with infinite population & finite population, multi-channel - poison arrivals - Exponential service times with infinite population.

Introduction to Optimization Techniques: Single objective & Multi objective optimization Techniques like G.A, NSGA, P.Q.O & MPSO Techniques.

Suggested Readings:

1. Hamdy, A. Taha, -Operations Research-An Introduction‖, Sixth Edition, Prentice Hall of India Pvt. Ltd.,1997
2. S.D. Sharma, Operations Research, Kedarnath, Ramnath & Co., Meerut,2009
3.J.B. Gupta, ―Utilization of Electric Power and Electric Traction‖ S.K. Kataria & Sons Publications, 2010 Hrvey M. Wagner, Principles of Operations Research, Second Edition,Prentice Hall of India Ltd., 1980.
4. V.K. Kapoor, Operations Research, S. Chand Publishers, New Delhi,2004 R. Paneer Selvam, Operations Research, Second Edition, PHI Learning Pvt. Ltd., New Delhi,2008.

Course Code	Course Title						Core/Elective
U23CD501, U23CD603, U24CD303	**Software Engineering**						**Core**
Prerequisite	**Hours Per Week**				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
OS	**3**	**-**	**-**	**-**	**40**	**60**	**3**

Course Objectives:

1. To introduce the basic concepts of software development- processes from defining a product to shipping and maintaining that product.
2. To impart knowledge on various phases, methodologies and practices of software development.
3. To understand the importance of testing in software development and study various testing strategies and software quality metrics.
4. To impart knowledge on various phases, methodologies and practices of software development
5. To understand the importance of testing in software development, study various testing strategies along with its relationship with software quality and metrics

Course Outcomes:

At the end of the course, student would be able to

1. Describe and compare alternative approaches and techniques used across various phases of the software development lifecycle.
2. Develop a complete software project independently by applying appropriate design principles, tools, and methodologies.
3. Identify and analyze the real-world challenges involved in developing large-scale software systems.
4. Design and construct software architecture independently or in a team by recognizing recurring problems and applying relevant design patterns.
5. Evaluate software product quality using appropriate metrics while addressing practical development challenges.

UNIT-I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). Process models: The waterfall model, Spiral model and Agile methodology

UNIT – II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT – III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT – IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT – V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards

Suggested Readings:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 8th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. Software Engineering by Dr Salman Abdul Moiz, Printed and published by All India Council for Technical

Education (AICTE), New Delhi.

Course Code	Course Title						Core/Elective
U24CS301	**Discrete Mathematics**						**Core**
Prerequisite	Hours Per Week				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
Basic Maths	**3**		_	_	**40**	**60**	**3**

Course Objectives:

Develop ability to

1. Understand concepts of Mathematical Logic, mechanisms of inference rules for propositional and predicate logic and their applications
2. Understand the concepts of Sets, Relations, Functions and their applications.
3. Learn the concepts of Algebraic Structures, basics of counting, Principles of inclusion/exclusion and the pigeonhole methodology.
4. Understand Generating Functions, Recurrence Relations and various ways of solving them.
5. Understand basic definitions and properties of graphs and their applications in computer science and engineering.

Course Outcomes:

At the end of the course, student would be able to

1. Distinguish between Propositional Logic and Predicate Logic, deriving valid proofs of inference and checking the validity of inferences.
2. Illustrate by examples the basic terminology of sets, relations, functions and algebraic structures along with their associated operations.
3. Demonstrate basics of counting, principles of permutations, combinations, applying inclusion/exclusion principle and the pigeonhole methodology in solving counting problems.
4. Demonstrate Generating functions, write recurrence relations and apply various techniques solving recurrence relations.
5. Transform a problem in computer science and engineering as a graph to solve it efficiently using concepts of graph theory.

UNIT-I

Mathematical Logic: Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Quantifiers, universal quantifiers. Predicates: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving.

UNIT-II

Relations: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Lattices, Hasse diagram. Functions: Inverse Function Composition of functions, recursive Functions, Lattice and its Properties.

UNIT-III

Elementary Combinatorics: Basis of counting, Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial Multinomial theorems, the principles of Inclusion – Exclusion. Pigeon hole principles and its application.

UNIT-IV

Recurrence Relation: Generating Functions, Function of Sequences Calculating Coefficient of generating function, Recurrence relations, Solving recurrence relation by substitution and Generating funds. Characteristics solution of inhomogeneous Recurrence Relation.

UNIT-V

Graph Theory: Basic Concepts, Representation of Graph, Isomorphism, Sub graphs, Spanning Trees, Planar Graphs, Multi graphs, Euler circuits, Hamiltonian graphs, Chromatic Numbers. Graph Theory and Applications.

Suggested Readings:

1. Discrete Mathematical Structures with Applications to Computer Science, J.P. Tremblay, R. Manohar, McGraw Hill education (India) Private Limited. (UNITS - I , II)
2. Discrete Mathematics for Computer Scientists & Mathematicians, Joe L. Mott, Abraham Kandel, Theodore P. Baker, Pearson, 2nd ed. (Units - III, IV, V)
3. Elements of Discrete Mathematics- A Computer Oriented Approach- C L Liu, D P Mohapatra Third Edition, Tata Mc Graw Hill.
4. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition.TMH.
5. Discrete Mathematical Structures Theory and Application-Malik & Sen, Cengage.

Course Code	Course Title						Core/Elective
U24CS302	**Data Structures**						**Core**
Prerequisite	Hours Per Week				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
PPS	**3**	-	_	_	**40**	**60**	3

Course Objectives:

Develop ability to

1. Develop skills to design and analyze simple linear and nonlinear data structures, such as stacks, queues and lists and their applications.
2. Gain programming skills to implement sorting and searching algorithms
3. Strengthen the ability to identify and apply the suitable data structures for the given real world problem.
4. Gain knowledge in practical applications of data structures
5. Understand essential for future programming and software engineering courses.

Course Outcomes:

At the end of the course, student would be able to

1. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given problem
2. Implement various data structures using arrays, linked lists
3. Develop ADT necessary for solving problems based on Stacks and Queues
4. Implement binary trees, general tree structures, advanced search trees, heaps, graphs.
5. Implement hash functions and handle collisions.

UNIT – I

Introduction Data Structures and Algorithms:

Introduction to Data Structures, classification of data structures, operations on data structures; Algorithm Specification, Recursive algorithms, Data Abstraction, Performance analysis- Time Complexity and Space Complexity, Asymptotic Notation-Big O, Omega, and Theta notations.

UNIT – II

Searching Techniques: Linear search, Binary Search algorithms.

Sorting Techniques: Bubble Sort, Insertion sort, Selection Sort, Merge Sort, and Quick Sort. Comparison among sorting techniques.

Stacks: Stack ADT, definition and operations, Implementations of stacks using array, applications of stacks, Arithmetic expression conversion and evaluation.

UNIT - III

Queues: Queue ADT, definition and operations, Implementation of queues using Arrays, applications of linear queue, circular queue and double ended queue (deque).

Linked Lists: Introduction, Single linked list, representation of a linked list in memory, operations on a Singly Linked List, Implementation of Singly Linked List.

UNIT – IV

Doubly Linked Lists: Operations on Doubly Linked List, Implementation of Doubly Linked List, Circular linked list, Implementation of Stack and Queue using linked list.

Hashing: Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation of Overflow Techniques

UNIT – V

Trees: Introduction, Binary Trees, Tree Traversals, Threaded Binary Trees, Binary Search Tree, Heap Tree, AVL Tree.

Graphs: Graph Abstract Data Type, Representation of Graph, Graph Traversals -DFS and BFS, Spanning Tree, Prim‘s and Kruskal‘s Algorithms.

Suggested Readings:

1. "Fundamentals of Data Structures in C", Ellis Horowitz, Sartaj Sahani, Susan Anderson Freed, Computer Science Press, 2007
2. S. Lipschutz, "Data Structures", Tata McGraw Hill Education, 1st Edition, 2014.
3. D. Samanta, "Classic Data Structures", PHI Learning, 2nd Edition, 2004.
4. Mark A Weiss, "Data Structures and Algorithm Analysis In C", Second Edition (2002), Pearson
5. "Data Structures and Algorithms in C++", Second Edition, 2011 by Michael T. Goodrich and Roberto Tamassia.

Course Code	Course Title						Core/Elective
U24IT301	**Database Management Systems**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
PPS	**3**	**-**	**-**	**-**	**40**	**60**	**3**

Course Objectives:

The objectives of this course is to impart knowledge

1. Understand the role of database management system in an organization and learn the database concepts.
2. Design databases using data modelling and Logical database design techniques.
3. Construct database queries using relational algebra and calculus and SQL.
4. Understand the concept of a database transaction and related concurrent, recovery facilities.
5. Understand the concepts of Triggers and Stored Procedures.

Course Outcomes

At the end of the Course, Student would be:

1. Design ER-models to represent simple database application scenarios and Construct database queries using SQL.
2. Construct database queries using relational algebra and calculus.
3. Recognize and identify the use of normalization and functional dependency in database design.
4. Apply the concept of a database transaction and related concurrent, recovery facilities.
5. Apply and relate how to evaluate a set of queries in query processing.

UNIT - I

Conceptual Modeling Introduction: Introduction to Data bases: Purpose of Database systems, view of data, data models, Database languages, Database users, various components of overall DBS architecture, various concepts of ER model, basics of Relational Model.

SQL Query – Basics:

SQL – Data Definition commands, Queries with various options, Data manipulation commands, Views, Joins, views, integrity and security.

UNIT – II

Relational Approach

Relational algebra and calculus: Relational algebra, selection and projection, set operations, renaming, joins, division, examples of algebra queries, relational calculus: Tuple relational calculus, Domain relational calculus, expressive power of algebra and calculus.

UNIT - III

Introduction to NoSQL: Introduction, Overview and History of NoSQL Databases – The Definition of the Four Types of No SQL Databases, differences between SQL and No SQL.

Normalization: Pitfalls of RDBD, Lossless join decomposition, functional dependencies, Armstrong axioms, normalization for relational databases 1st, 2nd and 3rd normal forms, BCNF, Basic definitions of MVDs and JDs, 4th and 5th normal forms.

UNIT - IV

Transaction Management: Transaction processing: Transaction concept, transaction State, implementation of atomicity and durability, concurrent executions, serializability, recoverability. Concurrency Control: Lock-based protocols, timestamp-based protocols, validation-based protocols, multiple granularities, multi-version schemes, deadlock handling.

UNIT – V

Data Storage: Overview of physical storage media, magnetic disks, storage access, file organization, organization of records in files.

Indexing and Hashing: Basic concepts, types of indexing, difference between B and B+ Indexing, static hashing, Dynamic Hashing.

Suggested Readings:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill, 6th Edition, 2017.
2. Ramez Elmasri, Shamkant B. Navathe, "Fundamental Database Systems", Pearson Education, 6th Edition, 2014.
3. Raghu Ramakrishnan, "Database Management System", Tata McGraw-Hill Publishing Company, 3rd Edition, 2007.
4. Hector Garcia Molina, Jeffrey D. Ullman, Jennifer Widom, "Database System Implementation", Pearson Education, United States, 1st Edition, 2000.
5. Peter Rob, Corlos Coronel, "Database System, Design, Implementation and Management", Thompson Learning Course Technology, 5th Edition, 2003.

Course Code	Course Title						Core
U24CS3L1	**Data Structures Lab**						**Core**
Prerequisite	Hours Per Week				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
PPS Lab	**-**	**-**	**-**	**3**	**25**	**50**	**1.5**

Course Objectives:

Develop ability to

1. Understand essential concepts of simple linear and nonlinear data structures.
2. Apply the suitable data structures for the given real-world problems.
3. Acquire knowledge in practical applications of data structures.
4. Provide solutions for various graphical concepts.
5. Analyze and implement programming skills to implement sorting and searching algorithms.

Course Outcomes:

At the end of the course, student would be able to

1. Perform Sorting and Searching and be able to justify which sorting and searching techniques is suitable
2. Apply stacks and queues in solving problems
3. Evaluate binary trees, general tree structures, advanced search trees, heaps, graphs.
4. Apply hash functions and handle collisions.
5. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given Problem.

List of Experiments:

1. Implementation of Binary Search and Linear Search.
2. Implementation of Selection, Merge, Quick, and InsertionSort.
3. Implementation of Stacks and Queues using Arrays.
4. Implementation of Infix to Postfix Conversion, Postfix ExpressionEvaluation.
5. Implementation of Circular Queue using Arrays.
6. Implementation of Singly Linked List
7. Implementation of Doubly Linked List.
8. Implementation of Circular Linked List.
9. Implementation of Stacks, Queues using Linked Lists.
10. Implementation of Binary Search Tree. (Insertion, Deletion, and Search operations)
11. Implementation of Tree Traversal on Binary Trees.
12. Implementation of AVL Trees.
13. Implementation of Traversal on Graphs.
14. Implementation of Prim's and Kruskal's Algorithm.

Suggested Readings:

1. S. Lipschutz, "Data Structures", Tata McGraw Hill Education, Revised Edition, 2014.
2. D. Samanta, "Classic Data Structures", PHI Learning, 2nd Edition, 2004.
3. Mark A Weiss, Data Structures and Algorithm Analysis In C, SecondEdition (2002), Pearson

Course Code	Course Title						Core/Elective
U24IT3L1	**Database Management Systems Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	**T**	**D**	**P**			
PPS	**-**	**-**	**-**	**3**	**25**	**50**	**1.5**

Course Objectives:

1. Introduce ER data model, database design and normalization
2. Learn SQL basics for data definition and data manipulation
3. To understand the basic concepts and the applications of database systems.
4. Be acquainted with the basics of transaction processing and concurrency control.
5. Learn the concepts of Views, Stored Procedure and Triggers.

Course Outcomes

After completing this course, the student will be able to:

1. Design database schema for a given application and apply normalization
2. Gather skills in using SQL commands for data definition and data manipulation.
3. Demonstrate creation and usage of Views and Stored Procedures using SQL.
4. Develop solutions for database applications using procedures, cursors and triggers
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modelling, designing, and implementing a DBMS.

List of Experiments:

Scenario: Product-Sales database: South wind database is a sample database used by Organization. The database contains the sales data for South wind Traders; it is foods export-import Company. Using this schema to demonstrate the how customer can choose and order products, how orders are placed and how those products get delivered to the customer. Products: This Entity will have all the products details where suppliers will supply products based on customers demand. Supplies: This Entity will supply the products demanded by the customers. Shippers: This Entity will take the orders from suppliers and deliver to customers. Employees: Employees will monitor the orders placed by customers. Invoices: This Entity will take care of billing process based on customer order. Etc... Identify some more entities and find out relationship between them. Product-sales the above process involves many steps like

1. Analyzing the problem and identifying the Entities and Relationships,
2. E-R Model
3. Relational Model
4. Normalization
5. Creating the database
6. Querying.

Experiment 1: E-R Model

Analyze and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like Foreign Key and constraints like NULL, NOT NULL, CHECK etc. Example to create for products, customers, suppliers, orders, ,
employees, order details, categories, among others. Students should submit E-R diagrams using the above tables.

Experiment 2: DDL

How to create tables, altering the database or tables, dropping tables if not required. You will also try truncate, rename commands etc. Data Definition Language (DDL) : create , alter, drop.

Experiment 3: DML

Data Manipulation Language Commands (DML) commands are used to for managing data within schema objects. Exercising the commands using DML: insert, delete, update on the following tables: products, customers, suppliers, orders, , employees, order details, categories.

- INSERT – insert data into a table.

● UPDATE – updates existing

Experiment 4: Querying

data within a table.

● DELETE – deletes single or all records from a table.

Data Query Language – Select Populate all the tables designed in experiment: 2 with appropriate data.

Practice queries on Aggregate functions like count, max, min, avg, sum Practice queries like nested queries/co-related queries using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, group by, having etc.

Joins: Join , Left Outer Join, Right Outer Join, Self Join

Experiment 5 : Querying(continued…)

Some example to practice the queries:

1. Display all the order details of given a customer.
2. Display all the products.
3. Get the highest sold product from given supplier ID
4. List all products grouped by category
5. List the products, whose products unit price is greater then all the products of average.
6. List Details of order and customer of each order
7. List the products which were sold in year 1997
8. Display the total amount for each order
9. Display Order Details for given an order ID
10. Order Details: product name and unit price for given order ID Exercising Simple to complex
11. Queries using joins, nested and co-related queries.

Experiment 6 : Programs on pl/sql

1. Write aPL/SQL program to swap two numbers.
2. Write aPL/SQL program to find the largest of three numbers
3. Write aPL/SQLprogramtofindthetotalandaverageof6subjectsanddisplaythegrade.
4. Write aPL/SQL program to find the sum of digits in a given number.
5. Write aPL/SQL program to display the number in reverse order.
6. Write a PL/ SQL program to check whether the given number is prime or not.
7. Write aPL/SQL program to find the factorial of a given number.

Experiment 7 : Stored Procedures :

1. Create a stored procedure, Alter and Drop a procedure, IN, OUT, IN & OUT parameters
2. Create a Procedure to display order details of given customer ID like ordered, order Date,
Required Date, Shipped Date
3. Create a procedure to accept a customer ID and display the customer order history (product name and how much quantity ordered for that particular product)
Ex: product name, Total quantity he/she ordered.
4. Create a procedure to display Ten Most Expensive Products Columns should be displayed
Product name & Unit price

Experiment 8: Views

1. Create a view to display the current product list which is available (not discontinued)
2. Create a view to display the products by category
3. Display product name, quantity Per Unit, units In Stock, Discontinued
4. Create a view as ―Invoices‖ to display all the information from order, customer, and shipper for each Order Details

Experiment 9: Triggers

Demonstrate Create Trigger, Alter Trigger, Drop Trigger, Row Level, Table Level triggers, Before Insert, After Insert, Before Update, After Update, Before Delete, After Delete

Experiment 10: Case study: Book Publishing Company

A publishing company produces scientific books on various subjects. The books are written by authors who specialize in one particular subject. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications.

A publication covers essentially one of the specialist subjects and is normally written by a single author. When writing a particular book, each author works with on editor, but may submit another work for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particular subject for the above case study, do the following:

a. Analyze the data required.
b. Normalize the attributes.

Create the logical data model using E-R diagrams.

Experiment 11: Case Study: General Hospital

A General Hospital consists of a number of specialized wards (such as Maternity, Pediatric, Oncology, etc). Each ward hosts a number of patients, who were admitted on the recommendation of their own GP and confirmed by a consultant employed by the Hospital. On admission, the personal details of every patient are recorded. A separate register is to be held to store the information of the tests undertaken and the results of a prescribed treatment. A number of tests may be conducted for each patient. Each patient is assigned to one leading consultant but may be examined by another doctor, if required. Doctors are specialists in some branch of medicine and may be leading consultants for a number of patients, not necessarily from the same ward. For the above case study

For the above case study, do the following.

a. Analyze the data required.
b. Normalize the attributes.

Create the logical data model using E-R diagrams.

Suggested Readings

1. Raghurama Krishnan, Johannes Gehrke, —Database Management Systems‖, Tata McGraw Hill, 3rd Edition, 2008.
2. Silberschatz, Korth, —Database System Concepts‖, McGraw Hill, V edition, 2005.
3. Rick F. Vander Lans, —Introduction to SQL‖, Pearson education, 2007.
4. B. Rosenzweig and E. Silvestrova, —Oracle PL/SQL‖, Pearson education, 2004.
5. Dr. P. S. Deshpande, —SQL & PL/SQL for Oracle 10g‖, Black Book, Dream Tech, 2006.
6. M. Mc Laughlin, —Oracle Database 11g PL/SQL Programming‖, TMH, 2017.

Course Code	Course Title						Core/Elective
U24CS3L3	**Data Visualization Lab**						**--**
Prerequisite	**Hours Per Week**				**CIE**	**SEE**	**Credits**
	L	**T**	**D**	**P**			
PYTHON	**-**	**-**	**-**	**3**	**25**	**50**	**1.5**

Course Objectives:

The objectives of this course are:

1. Effective use of Business Intelligence (BI) technology (Tableau) to apply data visualization.
2. To discern patterns and relationships in the data.
3. To build Dashboard applications.
4. To communicate the results clearly and concisely.
5. To be able to work with different formats of data sets

Course Outcomes:

On Successful completion of this course, students will be able to:

1. Understand how to import data into Tableau.
2. Understand Tableau concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties.
4. Create a Dashboard that links multiple visualizations.

Use graphical user interfaces to create Frames for providing solutions to real-world problems

List of Experiments:

1. Understanding Data, what is data, where to find data, and Creating Your First Visualization in Python.

2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts (line, bar charts, Tree maps), Using the Show me panel.

3. Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.

4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, and Formatting specific parts of the view.

5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.

6. Structuring your data, Sorting and filtering Tableau data, and Pivoting Tableau data.

7. Creating Dashboards & Storytelling, creating your first dashboard and Story, and Design for different displays, and publish your visualization.

8. Creating custom charts, cyclical data, circular area charts, and Dual Axis charts.

9. Advanced Visualization Tool Power BI: Using Filters, Using the Detail panel, using the Size panels, customizing filters, and customizing tooltips, and formatting your data with colors.

10. Visualization data supported chart types in Power BI, Map Visualizations, Color palettes in Charts, Loading Shapes, text boxes, and images.

11. Explore the Google Data Studio Dashboard and connect to your first source.

12. Create a Report on Google Data Studio and share the report.

Suggested Reading:

1. Data Visualization in Python by Daniel Nelson, 2020 Stack Abuse.

2. Tableau Your Data: Fast and Easy Visual Analysis with Tableau Software by Daniel G Murray, WILEY

3. Microsoft Power BI cookbook, Brett Powell, 2nd edition.

Course Code	Course Title						Core / Elective
U24CS3L2	C PROGRAMMING LAB						Core
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Computer Basics	L	T	D	P			
	-	-	-	2	50	-	-

Course Objectives:

The objectives of this course are:

1. To understand the fundamentals of programming in C language.
2. To write, compile and debug programs in C.
3. To formulate solution to problems and implement in C.
4. To effectively choose programming components to solve computing problems.

Course Outcomes:

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

1. Choose appropriate data type for implementing programs in C language.
2. Design and implement modular programs involving input output operations, decision making and looping constructs.
3. Implement search and sort operations on arrays.
4. To decompose a problem into functions and to develop modular reusable code.
5. Apply the concept of pointers for implementing programs on dynamic memory management and string handling.

List of Experiments

1. Finding roots of quadratic equation.
2. Finding maximum and minimum of given set of numbers.
3. Conversion of Binary to Decimal, Octal, Hexadecimal and vice versa.
4. Generating Pattern & Pyramid of Numbers
5. Recursion: factorial, Fibonacci, GCD.
6. Matrix addition and multiplication using arrays.
7. linear search and binary search using non recursive procedures
8. Bubble Sort and Selection Sort
9. Programs on Pointers: Pointers to Arrays, Pointer and Function
10. Functions for string manipulations.
11. Programs on Structures and Unions
12. Finding the number of characters, words and lines of given text file using file handling functions.

Suggested Readings:

1. “Theory and practice of Programming with C“, Byron Gottfried, Schaum's Outline,McGraw-Hill 1996.
2. “Computer Fundamentals and Programming in C”, A.K. Sharma, Universities Press, 2nd Edition, 2018.
3. “Programming in ANSI C”, E. Balaguruswamy, Tata McGraw-Hill Education,2008.
4. “The C Programming Language”, Brian W. Kernighan and Dennis M. Ritchie, Prentice Hall of Indian 1988.