# B.Tech. Programme in
# COMPUTER SCIENCE AND ENGINEERING

**(III semester course structure and syllabi)**


**2019 Regulations**

**GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING**

(AUTONOMOUS)

AFFILIATED TO JNTU- KAKINADA

MADHURAWADA, VISAKHAPATNAM

# Course structure for III semester

| COMPUTER SCIENCE AND ENGINEERING | | | | | |
|---|---|---|---|---|---|
| **Course Code** | **Name of the Course** | **L** | **T** | **P** | **C** |
| 19BM1105 | Mathematics-III (Discrete Mathematical Structures) | 3 | 0 | 0 | 3 |
| 19HM1101 | Accounting and Economics for Engineers | 3 | 0 | 0 | 3 |
| 19CT1103 | Data Structures and Algorithms | 3 | 0 | 0 | 3 |
| 19EC11D4 | Computer Organization | 3 | 0 | 0 | 3 |
| 19CT1104 | Database Management Systems | 3 | 0 | 0 | 3 |
| 19CT1105 | Database Management Systems Lab | 0 | 0 | 3 | 1.5 |
| 19CS1102 | Data Structures Lab | 1 | 0 | 3 | 2.5 |
| 19CT1106 | Python Programming Lab | 0 | 0 | 3 | 1.5 |
| 19HM11Z1 | Human Values and Professional Ethics | 3 | 0 | 0 | 0 |
| | **TOTAL** | **16** | **0** | **9** | **20.5** |

# DISCRETE MATHEMATICAL STRUCTURES

**(common to CSE & IT)**

**Course Code : 19BM1105**

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 0 | 3 |

**Course Objectives:**

- Familiarize the basics of mathematical logic.
- To introduce basics of group theory and its applications
- Bring awareness of basic concepts of graphs and its applications.

**Course Outcomes:**

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

**CO1**: verify the validity of a logical flow of arguments.

**CO2**: identify various types of relations and their properties.

**CO3**: solve recurrence relations of various types.

**CO4**: identify algebraic structures and learn modular arithmetic.

**CO5**: understand various concepts of graphs and spanning trees.

**Unit I: Mathematical logic** **10 Lectures**

Statements and notations, connectives, well formed formulas, tautologies, equivalence of formulas, duality law, tautological implications, other connectives, normal forms, rules of inference, consistency of premises and indirect method of proof, automatic theorem proving.

(Sections 1-1, 1-2, 1-2.1 to 1-2.4, 1-2.6 to 1-2.11, 1-2.14, 1-3, 1-3.1 to 1- 3.4, 1-4.2 to 1-4.4 of textbook 1)

**Learning Outcomes:**

At the end of this unit, the student will be able to

1. determine the equivalence of formulas and implement the logic for mathematical proofs (L3)
2. infer the consistency of an argument (L4)
3. test the validity of the conclusion based on the hypothesis (L5)

**Unit II: Relations** **8 Lectures**

Relations, properties of binary relations in a set, Relation matrix and Graph of a relation, partition and covering of a set, equivalence relations, compatibility relations, composition of binary relations.

(Sections 2-3.1 to 2-3.7 of the textbook 1)

**Learning Outcomes:**

At the end of this unit, the student will be able to

1. identify the different types of relations (L1)
2. classify the data into classes using equivalence relations (L4)
3. use the relation matrix to check equivalence relation (L3)

**Unit III: Recurrence relations** **9 Lectures**

Generating function of sequences, calculating coefficients of a generating function, recurrence relations, solving linear recurrence relations by substitution method, generating function method and method of characteristic roots, solving inhomogeneous recurrence relations.

(Sections 3.1- 3.6 of the text book 2)

**Learning Outcomes:**

At the end of this unit, the student will be able to

1. formulate recurrence relations of a sequence (L6)
2. solve linear recurrence relations (L3)
3. explain the method of solving inhomogeneous recurrence relations (L2)

**Unit IV: Algebraic structures** **10 Lectures**

Algebraic structure, group, abelian group, subgroup, ring, field- definitions and examples, residue arithmetic- applications of Chinese remainder theorem, Fermat's theorem, Euler's theorem.

(Sections 3-5.1 to 3-5.4, 3-6.2 of the textbook 1)

**Learning Outcomes:**

At the end of this unit, the student will be able to

1. classify an algebraic structure (L4)
2. discuss the basic properties of group
3. use residue arithmetic to computer applications (L3)

**Unit V: Graph theory** **10 Lectures**

Basic concepts of a graph, isomorphism and subgraph, tree and its properties, DFS, BFS algorithms for finding a spanning tree, Kruskal's and Prim's algorithms for finding a minimal spanning tree.

(Sections 5.1-5.4 of textbook 2)

**Learning Outcomes:**

At the end of this unit, the student will be able to

1. classify different graphs (L4)
2. determine a spanning tree from a graph (L3)
3. generate a minimal spanning tree from a graph (L6)

**Text books:**


1. J.P. Tremblay and R. Manohar, "*Discrete Mathematical Structures with Applications to Computer Science*", Tata McGraw Hill, 1997.
2. Joe L. Mott, Abraham Kandel and T. P. Baker, "*Discrete Mathematics for Computer Scientists & Mathematicians*" , 2nd edition, Prentice Hall of India Ltd, 2012.


**References:**


1. Keneth. H. Rosen, "*Discrete Mathematics and its Applications*", 6th edition, Tata McGraw-Hill, 2009.
2. Richard Johnsonburg, "*Discrete mathematics*", 7th edition, Pearson Education, 2008.
3. Narsingh Deo, "*Graph Theory with Applications to Engineering and Computer Science*", Prentice Hall of India, 2006.

# ACCOUNTING AND ECONOMICS FOR ENGINEERS

| Course Code: 19HM1101 | L | T | P | C |
|---|---|---|---|---|
| | 3 | 0 | 0 | 3 |

**Course Objective**

To provide an insight into the basic concepts of financial accounting and economic theories and to develop the managerial skills of the students, for effective business operations.

**Course Outcomes**

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

**CO 1:** understand the basic concepts of business and various forms of business organisations.
**CO 2**: understand the accounting system and prepare necessary books of accounts.
**CO 3:** analyse and interpret the financial statements using accounting ratios.
**CO 4:** describe the concepts of managerial economics and apply the concepts of demand analysis.
**CO 5:** describe various production theories, market structures and apply the concepts of break even techniques for managerial decisions.

**UNIT - I: Concepts and Forms of Business Organisation (10 Lectures)**

**Business:** Characteristics, Objectives, Classification of business activities - Industry and Commerce, Industry-types: primary, secondary, tertiary; Commerce-trade: types-internal, external; wholesale and retail and auxiliaries to trade.

**Forms and Features of Business Organisation:** Sole Proprietorship, Partnership - Definition, kinds of partners, Advantages and limitation of partnership firm, Partnership deed, Joint Stock Company - Concept, merits and limitations, Types, Formation of company, Public sector and private sector enterprises, Forms of public sector enterprises.

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

- Understand various business concepts and business activities. (L2)
- Outline various forms of business organisations. (L4)
- Describe the characteristics of sole proprietorship, partnership firm and Joint Stock Company. (L2)
- Explain the process of formation of a company. (L2)
- Explain various forms of public enterprises and their characteristics. (L2)

**UNIT – II: Introduction to Financial Accounting (10 Lectures)**

**Financial Accounting:** Definition, Importance, Principles - Concepts & Conventions, Double entry book keeping system, Bases of accounting - Cash basis and Accrual basis, Journal, Ledger, Subsidiary books and Trial Balance.

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

- Describe accounting concepts and conventions. (L2)
- Understand the double entry system of book keeping. (L2)
- Understand bases of accounting and their significance. (L2)
- Illustrate journal, ledger and various subsidiary books. (L4)
- Describe the purpose and preparation of trial balance. (L2)

**UNIT – III: Preparation and Interpretation of Financial Statements (10 Lectures)**

**Financial Statements:** Objective, Importance and Limitations, Trading Account, Profit and Loss Account, Balance Sheet- Grouping of assets and liabilities, Preparation of final accounts with simple adjustments.

**Interpretation of financial statements:** Accounting Ratios - Objectives, Classification, Limitations and Computation (simple numerical problems).

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

- Demonstrate the ability to develop Trading Account, Profit & Loss Account and Balance Sheet (L3)
- Understand the treatment of adjustments in final accounts (L2)
- Solve and interpret various accounting ratios (L3)

**UNIT – IV: Introduction to Managerial Economics and Demand Analysis (10 Lectures)**

**Managerial Economics:** Definition, Nature and Scope of Managerial Economics

**Demand Analysis:** Definition, types of demand, Demand Determinants, Law of Demand and its exceptions

**Elasticity of Demand:** Definition, Types, Significance of Elasticity of Demand

**Demand Forecasting:** Definition, methods of demand forecasting

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

- Describe the nature and scope of managerial economics (L2)
- List out the different types of demand (L1)
- Explain the law of demand and its exceptions (L2)
- Understand various types of elasticity of demand (L2)
- Outline various demand forecasting techniques (L4)

**UNIT - V: Theories of Production, Cost Analysis and Market Structures (10 Lectures)**

**Production Function:** Concept, Law of Variable Proportions, Iso-quants and Iso-costs and Least Cost Combination of Inputs.

**Cost Analysis**: Types of Cost (Short run and Long run, Fixed and Variable cost, Marginal cost, Opportunity cost and Replacement cost). Break-Even Analysis (BEA) - Determination of Break-Even Point (Simple numerical problems) - Managerial applications and limitations of BEA.

**Types of markets:** Perfect, Monopoly and Monopolistic markets – Concepts and Features.

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

- Explain the production function with one variable of input and two variables of input (L2)
- Describe various cost concepts (L2)
- Determine Break Even Point (L3)
- Solve managerial problems using BEA (L3)

**TEXT BOOKS**

1. SN Maheswari, SK Maheswari & SK Maheswari, Financial Accounting, 6th edition, Vikas Publications, 2018

2. M. Haneeef & A. Mukherjee, Financial Accounting, 2nd edition, Tata McGraw Hill, 2018
3. R.L. Varshney & K.L Maheswari, Managerial Economics, 19th edition, S. Chand Publishers, 2018
4. H L Bhatia & S N Maheshwari, Economics for Engineers, 3rd edition, Vikas Publications, 2017
5. A R Aryasri, Managerial Economics and Financial Analysis, 3rd edition, Tata McGraw Hill, 2012
6. S A Siddiqui & A. S. Siddiqui, Managerial Economics and Financial Analysis, 1st edition, New Age Publishers, 2015

**REFERENCES**

1. Ramachandran N, Ram Kumar Kakani, Financial Accounting for Management, 2nd Reprint edition, Tata McGraw Hill, 2006
2. SP Jain & KL Narang, Financial Accounting, 12th edition, Kalyani Publishers, 2014
3. NCERT, Class XI, 2019, Business Studies, -edition, NCERT Publication Division, 2018
4. P Venkata Rao & J. V.Prabhakar Rao, Managerial Economics & Financial Analysis, 1stedition, Maruti Publications, 2011
5. Dominick Salvatore, Managerial Economics: Principles and Worldwide Applications, 8thedition, Oxford University Press, 2012
6. D N Dwivedi, Managerial Economics, 8th edition, PHI Publication, 2010

# DATA STRUCTURES AND ALGORITHMS

(Common to CSE & IT)

**Course Code: 19CT1103**

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 0 | 3 |

**Prerequisites:** Problem solving Using C, Mathematics.

**Course Outcomes:**

At the end of the Course the student shall be able to

CO1: Analyze algorithms and describe searching, sorting and hashing techniques.
CO2: Describe the concepts of stacks and queues.
CO3: Apply the concepts of linked lists.
CO4: Describe the concepts of trees.
CO5: Explain the concepts of graphs.

**UNIT-1** **10 Lectures**

**Analysis of Algorithms:** Efficiency of algorithms, Apriori Analysis, Asymptotic notations, Time complexity of algorithms using O notation, Polynomial Vs Exponential algorithms, Average, Best, Worst case complexities, Analyzing recursive programs. **Searching:** Introduction, Linear search, Binary search, Fibonacci search. **Internal Sorting:** Introduction, Bubble sort, Insertion sort, Selection sort. **Hashing:** Introduction, Hash Table Structure, Hash Functions.

**Learning Outcomes:**
At the end of the module, students will be able to:
1. Illustrate how linear, binary search and Fibonacci search would work with examples. (L2)
2. Describe insertion, selection, and bubble sort. (L2)
3. Implement C programs for linear, binary, and Fibonacci search using arrays. (L3)
4. Describe the concepts of Hash functions. (L2)

**UNIT-II** **10 Lectures**

**Stacks:** Introduction, Stack operations, Applications. **Queues:** Introduction, Operations on queues, Circular queues, Priority queues, Applications.

**Learning Outcomes:**
At the end of the module, students will be able to:
1. Illustrate how stack and queue would work with example. (L2)
2. Describe the advantages of circular, priority queue. (L2)
3. Describe the applications of stack, queue, circular queue and priority queue.(L2)
4. Implement C programs for stack and queue using arrays. (L3)

**UNIT-III** **10 Lectures**

**Linked Lists:** Introduction, Singly linked lists, Circular linked lists, Doubly linked lists, Multiple linked lists, Applications. **Linked stacks and linked queues:** Introduction, Operations on linked stacks and linked queues, Dynamic memory management, Implementation of linked representations, Applications.

**Learning Outcomes:**

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

1. Describe the advantages of linked implementation over array implementation of various data structures. (L2)
2. Demonstrate how to declare structures to be used in simple linked lists, double linked lists, and circular linked lists. (L3)
3. Implement the algorithms for inserting, deleting, and searching in a simple linked list. (L3)

**UNIT-IV** **10 Lectures**

**Trees and Binary Trees:** Introduction, Trees: Definition and basic terminologies, Representation of trees. Binary trees: Basic terminologies and types, Representation of binary trees, Binary tree traversals, Applications. **Binary Search Trees and AVL trees:** Introduction, Binary search trees: Definition and operations, AVL Trees: Definition and operations, Applications.

**Learning Outcomes:**

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

1. Explain what is meant by a balanced binary tree and why it is important. (L2)
2. Describe AVL trees, what the definition of AVL property is and demonstrate the ability to identify diagrams of trees as to whether they have the AVL property.(L2)
3. Discuss with diagrams and algorithms for a single left rotation, single right rotation, double left rotation, and double right rotation in AVL trees.(L2)
4. Demonstrate how to declare structures to be used in binary trees. (L3)
5. Implement the algorithms for inserting, deleting, and searching for nodes in a binary tree. (L3)

**UNIT-V** **10 Lectures**

**Graphs:** Introduction, Definitions and basic terminologies, Representations of graphs, Graph traversals and Applications.

**Learning Outcomes:**

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

1. Discuss a basic search algorithm for graphs. (L2)
2. Define minimal spanning tree and discuss with diagrams, Prim's algorithm for finding the minimal spanning tree of a graph. (L2)
3. Define shortest path and discuss with diagrams, Dijkstra's algorithm for finding the shortest path from node x to node y of a graph. (L2)

**Textbook:**

1. Ellis Horowitz, Sartaj sahni, Dinesh Mehta, *Fundamentals of Data structures in C++*, 2nd edition, University Press (India) Pvt.Ltd, 2008.

**References:**

1. G.A.V PAI, *Data Structures and Algorithms, Concepts, Techniques and Applications*, Volume1,1st Edition, Tata McGraw-Hill, 2008.

2. Richard F. Gilberg & Behrouz A. Forouzan, *Data Structures, Pseudo code Approach with C*, 2nd Edition, Cengage Learning India Edition, 2007.

3. Langsam, M. J. Augenstein, A. M. Tanenbaum, *Data structures using C and C++*, 2nd Edition, PHI Education, 2008.

4. Sartaj Sahni, Ellis Horowitz, *Fundamentals of Data Structures in C*, 2nd Edition, Orientblackswan, 2010.

**Web Reference:**

1. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html.

# COMPUTER ORGANIZATION
# (Common to CSE and IT)

**Course Code: 19EC11D4**

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 0 | 3 |

Prerequisites: Switching Theory and Logic Design.

**Course Outcomes:** At the end of the course the student will be able to

**CO1:** Discuss basic structure and organization of computers
**CO2:** Apply fixed and floating point arithmetic algorithms
**CO3:** Explain micro operations and input/output organization
**CO4:** Illustrate pipeline and Parallel Processors
**CO5:** Demonstrate memory design and memory organizations

## UNIT-I — 10 Lectures

**Functional Blocks of a Computer**

CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU: registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.

Learning outcomes: At the end of this unit, the student will be able to

1. understand the basic principles of Computer Systems (L2)
2. summarize various addressing modes for a Processor (L2)
3. describe Instruction Execution cycle (L2)

## UNIT-II — 10 Lectures

**Data Representation**

Signed number representation, fixed and floating point representations, character representation. Computer arithmetic: integer addition and subtraction, ripple carry adder, carry look-ahead adder, multiplication-shift-and add, Booth multiplier, carry save multiplier, Division restoring and non-restoring techniques, Division Techniques, floating point arithmetic.

Learning outcomes: At the end of this unit, the student will be able to

1. understand various Data Representations (L2)
2. summarize various Arithmetic Algorithms for fixed point Representation (L2)
3. apply various Algorithms for floating Point Arithmetic (L3)

## UNIT-III — 10 Lectures

**CPU Control Unit Design**

Hardwired and microprogrammed design approaches, Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers: program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes, Interrupt driven I/O, I/O device interfaces Serial Communication, asynchronous data transfer.

Learning outcomes: At the end of this unit, the student will be able to

1. describe knowledge on micro programming (L2)
2. understand principles of I/O devices are accessed (L2)
3. understand various data Transfers (L2)

## UNIT-IV 10 Lectures

**Pipelining**

Basic concepts of pipelining, throughput and speedup, pipeline hazards.

**Parallel Processors**

Introduction to parallel processors, Concurrent access to memory and cache coherency, Flynns Classification.

Learning outcomes: At the end of this unit, the student will be able to

1. illustrate the concepts of pipelining (L3)
2. discuss concepts of Parallel Processors (L2)
3. understand concepts of Multiprocessors (L2)

## UNIT-V 10 Lectures

**Memory System Design**

Semiconductor memory technologies, Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Learning outcomes: At the end of this unit, the student will be able to

1. understand memory system design (L2)
2. demonstrate the concept of hierarchical memory organization (L3)
3. explain the concept of Cache Memory (L2)

## Text Books:

M. Moris Mano, *Computer Systems Architecture*, 3rd Edition, Pearson Education, 2007.

## Reference Books:

1. John P. Hayes, *Computer Architecture and Organization*, 3rd Edition, WCB/McGraw-Hill,1988.
2. William Stallings, *Computer Organization and Architecture: Designing for Performance*, 10th Edition, Pearson Education, 2015.
3. Vincent P. Heuring and Harry F. Jordan, *Computer System Design and Architecture*, 2nd Edition, Pearson Education, 2003.
4. David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 5th Edition, Elsevier, 2014.
5. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, *Computer Organization and Embedded Systems*, 6th Edition, McGraw Hill Higher Education, 2012.

***

# DATABASE MANAGEMENT SYSTEMS

(Common to CSE & IT)

**Course Code: 19CT1104**

| L | T | P | C |
|---|---|---|---|
| 3 | 0 | 0 | 3 |

**Course Outcomes**:

At the end of the Course the student shall be able to
CO1: Illustrate the DBMS architecture and model a database using ER diagram.
CO2: Solve queries using procedural and non-procedural languages.
CO3: Apply the normalization techniques to improve the database design.
CO4: Explain the processing and controlling the consequences of concurrent data access.
CO5: Demonstrate the storage, accessing and recovery mechanisms.
.

**UNIT-I** **10 Lectures**

History of Database Systems, Database System Applications, Database System vs File System – View of Data – Data Abstraction –Instances and Schemas – Data Models – The ER Model – Relational Model – Other Models – Database Languages – DDL, DML —Transaction Management – Database System Structure – Storage Manager – The Query Processor.
Database Design and ER Diagrams – Beyond ER Design Entities, Attributes and Entity sets – Relationships and Relationship Sets –Additional Features of ER Model – Concept Design with the ERModel – Conceptual Design for Large enterprises.

**Learning Outcomes:**
At the end of the module, students will be able to:
1. describe when to use files and when to use a DBMS.(L2)
2. explain how data can be stored and processed.(L2)
3. apply data modeling tools like Entity-Relationship Diagrams.(L3)

**UNIT-II** **10 Lectures**

Introduction to the Relational Model – Integrity Constraint Over Relations – Enforcing Integrity Constraints – Querying Relational Data – Logical Database Design – Introduction to Views – Destroying / Altering Tables and Views. Relational Algebra – Selection and Projection Set Operations – Renaming – Joins – Division – Relational Calculus –Tuple Relational Calculus– Domain Relational Calculus.

**Learning Outcomes:**
At the end of the module, students will be able to:
1. describe the data using relational model.(L2)
2. solve queries using relational algebra and calculus. (L3)
3. summarize what views are for and how to use them.(L2)

**UNIT-III** **8 Lectures**

Schema Refinement – Problems Caused by Redundancy – Decompositions – Problem related to Decomposition – Reasoning about FDS – FIRST, SECOND, THIRD Normal Forms – BCNF– Schema Refinement in Database Design – Multi Valued Dependencies – FOURTH Normal Form.

**Learning Outcomes:**
At the end of the module, students will be able to:

1. examine the anomalies in a database.(L3)
2. determine the keys for a given set of functional dependencies.(L3)
3. apply the normal forms to normalize the tables.(L3)

**UNIT-IV** **11 Lectures**

Transaction Concept- Simple Transaction Model-Storage Structure- Transaction State- Implementation of Atomicity and Durability, Isolation– Concurrent – Executions – Serializability- Recoverability – Implementation of Isolation-Transactions as SQL Statements. Concurrency Control: Lock – Based Protocols-Deadlock Handling– Timestamp Based Protocols- Validation Based Protocols-Multi Version Schemes - Insert, Delete and Predicate Operations– Multiple Granularity.

**Learning Outcomes:**
At the end of the module, students will be able to:

1. explain transactions and their properties.(L2)
2. summarize the anomalies that occur without ACID properties.(L2)
3. explain the locking protocols used to ensure Isolation.(L2)
4. explain concurrency control.(L2)

**UNIT-V** **11 Lectures**

Recovery System: Recovery and Atomicity – Log Based Recovery– Recovery with Concurrent Transactions – Buffer Management – Failure with Loss of Non-Volatile Storage-Advance Recovery Systems- ARIES.

Data on External Storage – Overview of Physical Storage Media - RAID - File Organization and Indexing - Data Dictionary Storage – Cluster Indexes, Primary and Secondary Indexes – Index Data Structures – Hash Based Indexing – Tree Base Indexing – B+ Trees: A Dynamic Index Structure.

**Learning Outcomes:**
At the end of the module, students will be able to:

1. demonstrate the logging techniques used to ensure Atomicity and Durability.(L2)
2. apply Recovery techniques used to recover from crashes.(L3)
3. summarize how different indexing techniques work.(L2)

## Text Books:

1. Raghurama Krishnan, Johannes Gehrke, *Database Management Systems*, 3rd Edition, TATA McGrawHill, 2008.

## References Books:

1. Silberschatz, Korth, *Database System Concepts*, 6th Edition, McGraw Hill, 2010.

2. C.J.Date, *Introduction to Database Systems*, 7th Edition, Pearson Education, 2002.

3. Peter Rob & Carlos Coronel, *Database Systems design, Implementation, and Management*, 7th Edition, Pearson Education, 2000.

4. Elmasri Navrate, *Fundamentals of Database Systems*, 5thEdition, Pearson Education, 2007.

# DATABASE MANAGEMENT SYSTEMS LAB

(Common to CSE & IT)

| Course Code: 19CT1105 | L | T | P | C |
|---|---|---|---|---|
| | 0 | 0 | 3 | 1.5 |

**Course Outcomes**:

At the end of the Course the student shall be able to
CO1: Apply data definitions and data manipulation commands.
CO2: Apply nested queries and subqueries.
CO3: Demonstrate database applications with joins and views.
CO4: Use functions, procedures and procedural extensions of databases.
CO5: Use cursors, triggers and exception handling mechanism.

**List of Experiments:**

1. Data Definition Commands, Data Manipulation Commands for inserting and deleting of data from Tables

2. Data Manipulation Commands for updating and retrieving of data from Tables and Transaction Control statements.

3. Basic functions like Numeric, String, Date and conversion functions.

4. Database Querying – Simple queries.

5. Queries using aggregate functions , GROUP BY and HAVING clauses.

6. Database Querying – Nested queries,Sub-queries.

7. Queries using Joins.

8. Queries using Views.

Programs using PL/SQL:

9. Procedures and Functions.

10. Implicit and Explicit Cursors.

11. Triggers.

12. Exception Handling.

**Case studies:**

Students shall form in groups at the beginning of the semester and perform the following steps by the end of the semester and submit a project.

13. Design a Database for any real life application using ER model and normalize it.

14. Connect the Database through any programming language.

15. Build real life database applications.

## Text Books:

1. Elmasri Navathe, *Fundamentals of Database Systems*, 7th Edition, Pearson Education, 2017.

## Reference Books:

1. C.J.Date, *Introduction to Database Systems*, 7th Edition, Pearson Education, 2002.

2. Peter Rob & Carlos Coronel, *Database Systems design, Implementation, and Management*, 7th Edition, Pearson Education, 2000.

# DATA STRUCTURES LAB

| Course Code: 19CS1102 | L | T | P | C |
|---|---|---|---|---|
| | 1 | 0 | 3 | 2.5 |

**Prerequisites:** Computer Programming

**Course outcomes:** At the end of the course, a student will be able to

CO1**:** Develop programs using recursive functions.
CO2: Implement stacks and queues.
CO3: Develop Programs for searching, sorting and hashing techniques.
CO4: Implement different types of trees.
CO5: Apply concepts of graphs.

**List of Programs:**

1. Write a program that uses recursive function to:

i) Compute factorial of a given number ii) Solve the towers of Hanoi problem iii) GCD

2. Write a program to implement the following search algorithms:

i) Linear Search ii) Binary Search iii) Fibonacci Search.

3. Write a program to implement the following sorting algorithms

i) Bubble Sort ii) Insertion Sort iii) Quick Sort iv) Merge Sort.

4. Write a program to implement different types of hash functions.

5. Write a program that implements the following data structures using arrays:

i) Stack ii) Queue.

6. Write a program to implement the following Stack applications

i) Factorial ii) Evaluations of postfix expression iii) number conversion.

7. Write a program to implement the following types of queues

i) Priority Queue ii) Circular Queue.

8. Write a program to implement the following types of Lists

i) Singly linked list ii) Doubly linked list.

9. Write a program to implement binary tree using arrays and to perform binary tree traversals

i) inorder ii) postorder iii) preorder.

10. Write a program to perform the following operations using linked lists:

i) Insert an element into a binary search tree.

ii) Delete an element from a binary search tree.

iii) Search for a key element in a binary search tree.

11. Write a program to perform the following operations using linked lists:

i) Insert an element into an AVL tree. ii) Delete an element from an AVL tree.

12. Write a program for the implementation of BFS and DFS for a given graph.

## Additional Programs/Beyond Syllabi:

1. Write a program to implement double stack.

2. Write a program to implement Dijkstra's algorithm for the single source shortest path problem.

3. Write a program to reverse a linked list.

4. Write a program to demonstrate working of generic linked list.

5. Write a program to check whether an expression consists of balanced parenthesis or not using stack.

6. Write a program to interchange two adjacent nodes in a circular linked list.

7. Write a program to implement Topological sorting technique.

## References:

1. Ellis Horowitz, Sartajsahni, Dinesh Mehta, *Fundamentals of Data structures in C++*, 2nd edition, University Press (India) Pvt.Ltd, 2008.

2. G A V PAI, *Data Structures and Algorithms, Concepts, Techniques and Applications*, Volume-1, 1st Edition, Tata McGraw-Hill, 2008.

3. Richard F. Gilberg & Behrouz A. Forouzan, *Data Structures, A Pseudo code Approach with C*, 2nd Edition, Cengage Learning India Edition, 2007.

# PYTHON PROGRAMMING LAB

(Common to all the branches)

**Course Code: 19CT1106**

| L | T | P | C |
|---|---|---|---|
| 0 | 0 | 3 | 1.5 |

Prerequisites: Problem solving using C, Mathematics.

**Course Outcomes:** At the end of the Course the student shall be able to
CO1: Know comprehensions, different Decision Making statements and Functions.
CO2: Understand various data types like lists, tuples, strings etc
CO3: Understand and summarize different file handling operations.
CO4: Interpret Object oriented programming in Python types.
CO5: Design GUI applications in Python.

**List of Experiments:**

**1. Input and output**

(a) Print the "Python" for 1, print "Day - 1" for 2. By changing the variable "look" for each statement.

(b) Create a variable "number" and assign an Integer to the number. Check the assigned Integer is "Positive" or "Negative".

(c) Write a program to find the largest element among three Numbers.

(d) Write a program to print the sum of all the even number in the range 1 - 50 and print the even sum.

(e) Write a program to display all prime numbers within an interval of 20 and 50.

**2. Variables and Functions**

a. Write a program to swap two numbers without using a temporary variable.

b. Write a program to define a function with multiple return values.

c. Write a program which creates an adder given a value (Use only lambda).

d. Write a program to define a function using default arguments.

## 3. Loops and conditionals

a. Write a program to print the following patterns using loop:

*

**

***

****

b. Write a program to print multiplication tables of 8, 15, 69.

## 4. Strings

a. Write a program to find the length of the string without using any library functions.

b. Write a program to check if two strings are anagrams or not.

c. Write a program to check if a substring is present in a given string or not.

## 5. Lists

a. Write a program to perform the given operations on a list:

i. add ii. insert iii. slicing

b. Write a program to perform any 5 built-in functions by taking any list.

c. Write a program to get a list of even numbers from a given list of numbers.(use only comprehensions)

d. Write a program to implement round robin. Note: This routine to take a variable number of sequences and return elements from them in round robin till each sequence is exhausted. If one of the input sequences is infinite, this is also infinite. e.g if input is [1,2,3], (4,5) -> yield 1,4,2,5,3 one after the other. Use exception control and comprehensions to write elegant code. Hint: This requires you to understand variable arguments, lists, list copy, comprehensions, iterators, generators, exception handling, control flow etc.

## 6. Tuples

a. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenate tuples.

b. Write a program to return the top 'n' most frequently occurring chars and their respective counts.

e.g. aaaaaabbbbcccc, 2 should return [(a 6) (b 4)]

## 7. Sets

a. Write a program to count the number of vowels in a string (No control flow allowed).

b. Write a program that displays which letters are present in both strings.

c. Write a program to sort given list of strings in the order of their vowel counts.

**8. Dictionaries**

a. Write a program to generate a dictionary that contains numbers (between 1 and n ) in the form of (x, x*x).

b. Write a program to check if a given key exists in a dictionary or not.

c. Write a program to add a new key-value pair to an existing dictionary.

d. Write a program to sum all the items in a given dictionary.

**9. Files**

a. Write a program to sort words in a file and put them in another file. The output file should have only lower case words, so any upper case words from source must be lowered. (Handle exceptions)

b. Write a program to find the most frequent words in a text.(read from a text file)

**10. Classes**

a. Write a Python class named Person with attributes name, age, weight (kgs), height (ft) and takes them through the constructor and exposes a method get_bmi_result() which returns one of "underweight", "healthy", "obese".

b. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

**11. Arrays**

a. Write a program to create, display, append, insert and reverse the order of the items in the array.

b. Write a program to add, transpose and multiply two matrices.

**12. GUI**

Write a program to create a basic calculator using tkinter.

## Additional Programs:

1. Write a program to check whether a given number has an even number of 1's in its binary representation (No control flow allowed).

2. Write a program to implement user defined map() function.

3. Write a program to return a list in which duplicates are removed and the items are sorted from a given input list of strings.

4. Write a program to implement left binary search.

5. Write a program to change days to hours, hours to minutes, and minutes to seconds using currying of composition of function.

6. Write a program to generate an infinite number of even numbers (Use generator)

7. Write a program to convert a given iterable into a list. (Using iterator)

8. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.

9. Given a string and a number k, find the $k^{th}$ non-repeating character in the string. Consider a large input string with lakhs of characters and a small character set. How to find the character by only doing only one traversal of input string?

e.g. input : str = hellopython k=3; output = y

10. To add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

e.g. input : 'abc' output : 'abcing' ; input : 'string' output : 'stringly'

## References:


1. Y. Daniel Liang, *Introduction to programming using Python,* Pearson Publications, 2017.

2. Sheetal Taneja, *Python Programming A Modular Approach*, 1st edition, Pearson Publications, 2017.

3. Brett Slatkin (C), *Effective Python: 59 Specific Ways to Write Better Python,* I/C, 1st edition, Pearson Publications, 2015.

4. Ashok Namdev Kamathane and Amit Ashok Kamathane, *Programming and Problem Solving with Python* , 1st edition, McGraw Hill Education (India) Private Limited, 2017.

# HUMAN VALUES & PROFESSIONAL ETHICS

**Course Code: 19HM11Z1**

| L | P | T | C |
|---|---|---|---|
| 3 | 0 | 0 | 0 |

**Course Objective**

To impart the basic concepts of human values and ethical matters in professional life in order to make the student more effective in assessing humanitarian issues and making appropriate decisions.

**Course Outcomes:**
At the end of the course, the student will be able to:
**CO1:** Understand various concepts of ethics and ethical issues
**CO2:** Describe various theories relating to professional ethics at work place
**CO3:** Determine the fundamental concepts of social experimentation and problem solving
**CO4:** Understand an engineer responsibility for social safety and concepts of risk benefits
**CO5:** Describe human values and environment in the era of digitisation and globalisation of workplace.

**UNIT I: HUMAN VALUES: (10 lectures)**
Morals, Values and Ethics – Integrity-Work Ethics -Service learning – Civic Virtue – Respect for others – Living Peacefully –Caring –Sharing –Honesty – Courage–Cooperation–Commitment – Empathy –Self Confidence Character –Spirituality

**Learning Outcomes:**
After completion of this unit student will be able to:
- Define morals, values & work ethics. (L1)
- Demonstrate respecting others and developing civic virtue. (L3)
- Describe commitment (L1)
- Describe how to live peacefully (L1)

**UNIT II: ENGINEERING ETHICS: (10 lectures)**
Senses of Engineering Ethics-Variety of moral issued –Types of inquiry –Moral dilemmas –Moral autonomy –Kohlberg's theory-Gilligan's theory-Consensus and controversy –Models of professional roles-Theories about right action-Self interest -Customs and religion –Uses of Ethical theories –Valuing time –Cooperation – Commitment

**Learning Outcomes:**
After completion of this unit student will be able to:
- Summarise ethical responsibilities of the engineers. (L2)
- Describe various theories of professional ethics. (L2)
- Determine time management (L3)
- Recite different professional roles and theories. (L1)

**UNIT III: ENGINEERING AS SOCIAL EXPERIMENTATION (10 lectures)**

Engineering As Social Experimentation –Framing the problem –Determining the facts –Codes of Ethics – Clarifying Concepts –Application issues –Common Ground -General Principles –Utilitarian thinking respect for persons

.

**Learning Outcomes:**

After completion of this unit student will be able to:

- Describe issues relating to social experimentation. (L3)
- Determine the process of framing the problem and the facts. (L2)
- Summarise the concept of code of ethics. (L2)
- Demonstrate the concept of utilitarian thinking (L3)

**UNIT IV: ENGINEERS RESPONSIBILITY FOR SAFETY AND RISK (10 lectures)**

Safety and Risk –Assessment of safety and risk –Risk benefit analysis and reducing risk-Safety and the Engineer-Designing for the safety-Intellectual Property Rights (IPR)

**Learning Outcomes:**

After completion of this unit student will be able to:

- Define safety, risk & risk benefit analysis. (L1)
- Describe engineer's responsibility for providing safety. (L1)
- Summarise Intellectual Property Rights. (L2)

**UINIT V: GLOBAL ISSUES (10 lectures)**

Globalization –Cross culture issues-Environmental Ethics –Computer Ethics –Computers as the instrument of Unethical behaviour –Computers as the object of Unethical acts –Autonomous Computers-Computer codes of Ethics –Weapons Development -Ethics and Research –Analyzing Ethical Problems in research

**Learning Outcomes:**

After completion of this unit student will be able to:

- Interpret changes in human value system in the era of globalisation. (L2)
- Understand the computer ethics and environmental ethics (L2)
- Outline ethical issues relating to weapons development. (L4)
- Describe ethical problems in research. (L2)

**TEXT BOOKS:**


1. "Professional Ethics and Human Values" by A.Alavudeen, R.Kalil Rahman and M.Jayakumaran-LaxmiPublications.
2. "Engineering Ethics includes Human Values" by M.Govindarajan, S.Natarajananad, V.S.Senthil Kumar-PHI Learning Pvt. Ltd-2009
3. "Engineering Ethics" by Harris, Pritchard and Rabins, CENGAGE Learning, India Edition, 2009.
4. "Ethics in Engineering" by Mike W. Martin and Roland Schinzinger –Tata McGraw-Hill–2003.
5. "Professional Ethics and Morals" by Prof.A.R.Aryasri, Dharanikota Suyodhana-Maruthi Publications.
6. "Professional Ethics and Human Values" by Prof.D.R.Kiran-
7. "Indian Culture, Values and Professional Ethics" by PSR Murthy-BS Publications.