Jawaharlal Nehru Technological University Anantapur

(Established by Govt. of A.P., Act. No. 30 of 2008)

Ananthapuramu–515 002 (A.P) India

II year B.Tech

Course Structures and Syllabi under R19 Regulations

JNTUA Curriculum
INFORMATION TECHNOLOGY B. Tech Course Structure

2nd Year Course Structure

Semester – 3 (Theory - 6, Lab - 3)					
S.No	**Course No**	**Course Name**	**Category**	**L-T-P**	**Credits**
1.	19A54303	Mathematical Foundations of Computer Science	BS	3-0-0	3
2.	19A05301	Digital Logic Design	PC	3-0-0	3
3.	19A99304	Design Thinking	ES	2-0-0	2
4.	19A05302T	Database Management Systems	PC	3-0-0	3
5.	19A05303T	Object Oriented Programming Through Java	PC	3-0-0	3
6.	19A05402T	Design and Analysis of Algorithms	PC	2-1-0	3
7.	19A05302P	Database Management Systems Lab	PC	0-0-3	1.5
8.	19A05303P	Object Oriented Programming Through Java Lab	PC	0-0-3	1.5
9.	19A05402P	Design and Analysis of Algorithms Lab	PC	0-0-3	1.5
10.	19A99302	Biology For Engineers	MC	3-0-0	0
				Total	**21.5**

Semester - 4 (Theory - 6, Lab - 2)					
S.No	**Course No**	**Course Name**	**Category**	**L-T-P**	**Credits**
1.	19A54401	Number Theory and Applications	BS	3-0-0	3
2.	19A05401	Computer Organization	PC	3-0-0	3
3.	19A12401	Formal Languages and Automata Theory	PC	3-0-0	3
4.	19A05404T	Software Engineering	PC	3-0-0	3
5.	19A05304T	Python Programming	PC	2-1-0	3
6.	19A05403T	Operating Systems	PC	3-0-0	3
7.	19A05304P	Python Programming Lab	PC	0-0-3	1.5
8.	19A05403P	Operating Systems Lab	PC	0-0-3	1.5
9.	19A99301	Environmental Science	MC	3-0-0	0
				Total	**21**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
3	0	0	3

19A54303 MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Course Objectives

- To explain about the Boolean Algebra, Graph theory and Recurrence relations.
- To demonstrate the application of basic methods of discrete mathematics in Computer Science problem solving.
- To elucidate solving mathematical problems from algorithmic perspective.
- To introduce the mathematical concepts which will be useful to study advanced courses Design and Analysis of Algorithms, Theory of Computation, Cryptography and Software Engineering etc.
- To reveal how solutions of graph theory can be applied to computer science problems

UNIT- I

Statements and Notation, **Connectives**- Negation, Conjunction, Disjunction, Conditional and Bi-conditional, Statement formulas and Truth Tables. Well-formed formulas, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications.
Normal Forms: Disjunctive Normal Forms, Conjunctive Normal Forms, Principal Disjunctive Normal Forms (PDNF), Principal Conjunctive Normal Forms (PCNF), Ordering and Uniqueness of Normal Forms.
The Theory of Inference for the Statement Calculus: Rules of Inference, Consistency of Premises and Indirect Method of Proof.
The predicate Calculus, **Inference theory of the Predicate Calculus**.

Unit Outcomes:

- Describe logical sentences in terms of predicates, quantifiers, and logical connectives (L1).
- Evaluate basic logic statements using truth tables and the properties of logic (L5).
- Apply rules of inference to test the consistency of premises and validity of arguments (L3).
- Verify the equivalence of two formulas and their duals (L4).
- Find the Principal Conjunctive and Principal Disjunctive Normal Forms of a statement formula (L1).

UNIT-II

Set Theory: Basic concepts of Set Theory, Representation of Discrete structures, Relations and Ordering, Functions, Recursion.

Lattices and Boolean algebra: Lattices as Partially Ordered Sets, Boolean algebra, Boolean Functions, Representation and Minimization of Boolean Functions.

Algebraic Structures: Algebraic Systems: Examples and General Properties, Semi Groups and Monoids, Groups.

Unit Outcomes:

- Describe equivalence, partial order and compatible relations (L1).
- Compute Maximal Compatibility Blocks (L3).
- Identify the properties of Lattices (L2).
- Evaluate Boolean functions and simplify expression using the properties of Boolean algebra (L5).
- Infer Homomorphism and Isomorphism (L4).
- Describe the properties of Semi groups, Monoids and Groups (L1).

UNIT-III

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with repetitions, Enumerating Permutations and Combinations with constrained Representations, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion and Exclusion.

Unit Outcomes:

- Explain fundamental principle of counting (L2).
- Examine the relation between permutation and combination (L4).
- Solve counting problems by applying elementary counting techniques using the product and sum rules (L3).
- Apply permutations, combinations, the pigeon-hole principle, and binomial expansion to solve counting problems (L3).

UNIT-IV:

Recurrence Relations: Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, The method of Characteristic Roots, Solution of Inhomogeneous Recurrence Relations.

Unit Outcomes:

- Find the generating functions for a sequence (L1).
- Design recurrence relations using the divide-and-conquer algorithm (L6).
- Solve linear recurrence relations using method of Characteristic Roots (L3).
- Outline the general solution of homogeneous or Inhomogeneous Recurrence Relations using substitution and method of generating functions (L2).
- Solve problems using recurrence relations and recursion to analyze complexity of algorithms (L3).

UNIT-V:

Graphs: Basic Concepts, Isomorphism and Sub graphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi graphs and Euler Circuits, Hamiltonian Graphs, Chromatics Number, The Four-Color Problem

Unit Outcomes:

- Investigate if a given graph is simple or a multi graph, directed or undirected, cyclic oracyclic (L4).
- Describe complete graph and complete bipartite graphs (L1).
- Identify Euler Graphs, Hamilton Graph and Chromatic Number of a graph (L2).
- Apply the concepts of functions to identify the Isomorphic Graphs (L3).
- Apply depth-first and breadth-first search (L3).
- Apply Prim's and Kruskal's algorithms to find a minimum spanning tree (L3).

Course Outcomes:

After completion of this course the student would be able to

- Evaluate elementary mathematical arguments and identify fallacious reasoning (L5).
- Understand the properties of Compatibility, Equivalence and Partial Ordering relations, Lattices and Hassee Diagrams (L1).
- Understand the general properties of Algebric Systems, Semi Groups, Monoids and Groups (L1).
- Design solutions for problems using breadth first and depth first search techniques (L6)
- Solve the homogeneous and non-homogeneous recurrence relations (L3).
- Apply the concepts of functions to identify the Isomorphic Graphs (L2).
- Identify Euler Graphs, Hamilton Graph and Chromatic Number of a graph (L2).

Text Books:

1. Joe L. Mott. Abraham Kandel and Theodore P.Baker, "Discrete Mathematics for Computer Scientists & Mathematicians", 2nd Edition, Pearson, 2008 (For Units III to V).
2. J P Trembly and R Manohar, "Discrete Mathematical Structures with Applications to Computer Science", 1st Edition, McGraw Hill, 2017(For Unit I&II).

Reference Books:

1. Ralph P. Grimaldi and B.V. Ramana, "Discrete and Combinatorial Mathematics, an Applied Introduction", 5th Edition, Pearson, 2016.
2. Narsingh Deo, "Graph Theory with Applications to Engineering", Prentice Hall, 1979.
3. D.S. Malik and M.K. Sen "Discrete Mathematics theory and Applications", 1st Edition, Cenegage Learning, 2012.
4. C L Liu and D P Mohapatra, "Elements of Discrete Mathematics, A computer Oriented approach by", 4th edition, MCGRAW-HILL, 2018.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
3	0	0	3

19A05301 DIGITAL LOGIC DESIGN

(Common to CSE & IT)

Course Objectives:

- Understanding basic number systems, codes and logical gates.
- Acquiring the skills to manipulate and examine Boolean algebraic expressions, logical operations, and Boolean functions
- Acquainting with classical hardware design for both combinational and sequential logic circuits
- Experiencing about synchronous circuits.
- Obtaining the knowledge about various types of memories.

UNIT - I

Digital Systems and Binary Numbers: Digital Systems, Binary Numbers, Number base conversions, Octal, Hexadecimal and other base numbers, complements, signed binary numbers, binary codes, binary storage and registers, binary logic.

Boolean algebra and logic gates: Basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, Digital Logic Gates.

Unit Outcomes:

Student is able to

- Summarize the binary number system
- Illustrate various binary codes
- Describe the basic postulates of Boolean Algebra
- Develop a logic diagram using gates from a Boolean function

UNIT - II

Gate–Level Minimization: The Map Method, Four-Variable K-Map, sum of products, product of sums simplification, Don't care conditions, Simplification by Quine- McClusky Method**,** NAND and NOR implementation and other two level implementations, Exclusive-OR function.

Unit Outcomes:

Student is able to

- Apply the map method for simplifying Boolean Expressions.
- Apply Don't care conditions to simplify a Karnaugh map.
- Design two-level Boolean functions with NAND gates and NOR gates

UNIT - III

Combinational Logic: Combinational Circuits, Analysis of Combinational Circuits, Design Procedure, Binary Adder-Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers and Demultiplexers.

Unit Outcomes:

Student is able to

- Select fundamental combinational logic circuits.
- Analyze and design combinational circuits.
- Design Boolean function with a multiplexer.

UNIT - IV

Synchronous Sequential Circuits: Latches, Flip-flops, analysis of clocked sequential circuits, **Register and Counters:** Registers, Shift registers, Ripple counters, Synchronous counters and other counters.

Unit Outcomes:

Student is able to

- Explain the functionalities of latch and different flip-flops.
- Analyze and design clocked sequential circuits.
- Describe the use of sequential circuit components in complex digital systems.

UNIT - V

Memory and Programmable Logic: Random-Access memory, Memory decoding, ROM, Programmable Logic Array, Programmable Array Logic, Sequential programmable devices.

Digital Integrated Circuits: RTL and DTL Circuits, Transistor-Transistor Logic (TTL), Emitter-Coupled Logic (ECL), MOS, CMOS Logic, Comparisons of Logic Families

Unit Outcomes:

Student is able to

- Interpret the types of memories.
- Construct the Boolean functions with PLA and PAL.
- Describe the most common integrated circuit digital logic families.

Course Outcomes:

Students should be able to

- Analyze the number systems and codes.
- Decide the Boolean expressions using Minimization methods.
- Design the sequential and combinational circuits.
- Apply state reduction methods to solve sequential circuits.
- Describe various types of memories.

Text Books:

1. M. Morris Mano, M.D.Ciletti, “Digital Design”, 5th edition, Pearson, 2018.

Reference Books:

1. Donald P Leach, Albert Paul Malvino, Goutam Saha, “Digital Principles and applications”, Mc Graw Hill , 8th Edition,2015.
2. David J. Comer, “Digital Logic & State Machine Design”, Oxford University Press, 3rd Reprinted Indian Edition, 2012
3. R.D. Sudhakar Samuel, “Digital Logic Design”, Elsevier Publishers.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
2	0	0	2

19A99304 DESIGN THINKING

(Common to CSE & IT)

Preamble: Design is a realization of a concept or idea into a configuration, drawing or a product. Design thinking is cognitive and practical processes by which design concepts are developed by designers. Innovation is a new idea or a new concept. Product development is the creation of a new or different product that offers new benefits to the end user. This course introduces the design thinking in product innovation.

Course Objectives:

- To familiarize product design process
- To introduce the basics of design thinking
- To bring awareness on idea generation
- To familiarize the role of design thinking in services design

Unit -I

Introduction to design, characteristics of successful product development, product development process, identification of opportunities, product planning, Innovation in product development.

Unit -II

Design thinking: Introduction, Principles, the process, Innovation in design thinking, benefits of Design thinking, design thinking and innovation, case studies.

Unit-III

Idea generation: Introduction, techniques, Conventional methods, Intuitive methods, Brainstorming, Gallery method, Delphi method, Synecticsetc

Select ideas from ideation methods, case studies.

Unit-IV

Design Thinking in Information Technology, Design thinking in Business process model, Design thinking for agile software development, virtual collaboration, multi user and multi account interaction, need for communication, TILES toolkit, Cloud implementation.

Unit V

Design thinking for service design: How to design a service, Principles of service design, Benefits of service design, Service blueprint, Design strategy, organization, principles for information design, principles of technology for service design.

Course Outcomes:

Student should be able to

1. Generate and develop different design ideas.
2. Appreciate the innovation and benefits of design thinking.
3. Experience the design thinking process in IT and agile software development.
4. Understand design techniques related to variety of software services

Reference Books:

1. Christoph Meinel and Larry Leifer, “Design Thinking”, Springer, 2011
2. AdersRiiseMaehlum “Extending the TILES Toolkit” from Ideation to Prototyping
3. http://www.algarytm.com/it-executives-guide-to-design-thinking:e-book.
4. Marc stickdorn and Jacob Schneider, “This is Service Design Thinking”, Wiely, 2011
5. Pahl and Vietz, “Engineering Design”, Springer, 2007

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
3	0	0	3

19A05302T DATABASE MANAGEMENT SYSTEMS

(COMMON TO CSE & IT)

Course Objectives:

This course is designed to:

1. Train in the fundamental concepts of database management systems, database modeling and design, SQL, PL/SQL and system implementation techniques.
2. Enable students to model ER diagram for any customized application
3. Inducting appropriate strategies for optimization of queries.
4. Provide knowledge on concurrency techniques
5. Demonstrate the organization of Databases

UNIT-I: Introduction: Database systems applications, Purpose of Database Systems, view of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database users and Administrators,

Introduction to Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Operations

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

- ➢ Distinguish between Database and File System
- ➢ Categorize different kinds of data models
- ➢ Define functional components of DBMS

UNIT-II: Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub-queries, Modification of the Database. Intermediate SQL**:** Joint Expressions, Views, Transactions, Integrity Constraints, SQL Data types and schemas, Authorization.

Advanced SQL: Accessing SQL from a Programming Language, Functions and Procedures, Triggers, Recursive Queries, OLAP, Formal relational query languages.

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

- ➢ Outline the elements of the relational model such as domain, attribute , tuple, relation and entity
- ➢ Distinguish between various kinds of constraints like domain, key and integrity
- ➢ Define relational schema
- ➢ Develop queries using Relational Algebra and SQL
- ➢ Perform DML operations on databases

UNIT-III: Database Design and the E-R Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues.

Relational Database Design:
Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional-Dependency Theory, Algorithms for Decomposition, Decomposition Using Multivalued Dependencies, More Normal Forms

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

- ➢ Develop E-R model for the given problem
- ➢ Derive tables from E-R diagrams
- ➢ Differentiate between various normal forms based on functional dependency
- ➢ Apply normalization techniques to eliminate redundancy

UNIT-IV: Query Processing: Overview, Measures of Query cost, Selection operation, sorting, Join Operation, other operations, Evaluation of Expressions.
Query optimization: Overview, Transformation of Relational Expressions, Estimating statistics of Expression results, Choice of Evaluation Plans, Materialized views, Advanced Topics in Query Optimization.

At the end of the Unit, students will be able to:
1. Identify variety of methods for effective processing of given queries.
2. Obtain knowledge related to optimization techniques.

UNIT V: Transaction Management:
Transactions: Concept, A Simple Transactional Model, Storage Structures, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels, Transactions as SQL Statements.

Concurrency Control: Lock based Protocols, Deadlock Handling, Multiple granularity, Timestamp based Protocols, Validation based Protocols.
Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Nonvolatile Storage, Early Lock Release and Logical Undo Operations.

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

1. Understand various properties of transaction.
2. Design atomic transactions for an application.
3. Gain the knowledge about log mechanism and check pointing techniques for system recovery.

Course Outcomes

Students will be able to :

1. Design a database for a real world information system
2. Define transactions which preserve the integrity of the database
3. Generate tables for a database
4. Organize the data to prevent redundancy
5. Pose queries to retrieve the information from database.

TEXT BOOKS:

1. A. Silberschatz, H.F.Korth, S.Sudarshan, “Database System Concepts”, 6/e, TMH 2019

REFERENCE BOOKS:

1. Database Management System, 6/e RamezElmasri, Shamkant B. Navathe, PEA
2. Database Principles Fundamentals of Design Implementation and Management, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning.
3. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
3	0	0	3

19A05303T OBJECT ORIENTED PROGRAMMING THROUGH JAVA

(Common to CSE & IT)

Course Objectives:

- To understand object oriented concepts and problem solving techniques
- To obtain knowledge about the principles of inheritance and polymorphism
- To implement the concept of packages, interfaces, exception handling and concurrency mechanism.
- To design the GUIs using applets and swing controls.
- To understand the Java Database Connectivity Architecture

UNIT - I

Introduction: Introduction to Object Oriented Programming, The History and Evolution of Java, Introduction to Classes, Objects, Methods, Constructors, this keyword, Garbage Collection, Data Types, Variables, Type Conversion and Casting, Arrays, Operators, Control Statements, Method Overloading, Constructor Overloading, Parameter Passing, Recursion, String Class and String handling methods.

Unit Outcomes:

Student should be able to

- Understand the syntax, semantics and features of Java Programming Language.
- Learn object oriented features and understanding type conversion and casting.
- Understand different types of string handling functions and its usage.

UNIT - II

Inheritance: Basics, Using Super, Creating Multilevel hierarchy, Method overriding, Dynamic Method Dispatch, Using Abstract classes, Using final with inheritance, Object class,

Packages: Basics, Finding packages and CLASSPATH, Access Protection, Importing packages.

Interfaces: Definition, Implementing Interfaces, Extending Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces.

Unit Outcomes:

Student should be able to

- Implement types of Inheritance and developing new classes based on existing classes
- Distinguish between system packages and user defined packages.
- Demonstrate features of interfaces to implement multiple inheritances.

UNIT - III

Exception handling - Fundamentals, Exception types, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, The Console class, Serialization, Enumerations, Autoboxing, Generics.

Unit Outcomes:

Student should be able to

- Learn what exceptions are and how they are handled.
- Learn when to use exception handling and how to create user defined exceptions
- Learn the difference between various files and streams.

UNIT - IV

Multithreading: The Java thread model, Creating threads, Thread priorities, Synchronizing threads, Interthread communication.

The Collections Framework (java.util): Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Hashtable, Properties, Stack, Vector, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner.

Unit Outcomes:

Student should be able to

- Understand concurrency, parallelism and multithreading
- Learn the importance of collections and use prebuilt generic data structures from framework.

UNIT – V

Applet: Basics, Architecture, Applet Skeleton, requesting repainting, using the status window, passing parameters to applets

GUI Programming with Swings – The origin and design philosophy of swing, components and containers, layout managers, event handling, using a push button, jtextfield, jlabel and image icon, the swing buttons, jtext field, jscrollpane, jlist, jcombobox, trees, jtable, An overview of jmenubar, jmenu and jmenuitem, creating a main menu, showmessagedialog, showconfirmdialog, showinputdialog, showoptiondialog, jdialog, create a modeless dialog.

Accessing Databases with JDBC:

Types of Drivers, JDBC Architecture, JDBC classes and Interfaces, Basic steps in developing JDBC applications, Creating a new database and table with JDBC.

Unit Outcomes:

Student should be able to

- Learn how to use the Nimbus look-and-feel
- Understand the GUI programming.
- Understand basic steps in developing JDBC applications,

Course Outcomes:

After the completion of the course the student will be able

- To solve real world problems using OOP techniques.
- To apply code reusability through inheritance, packages and interfaces
- To solve problems using java collection framework and I/O classes.
- To develop applications by using parallel streams for better performance.
- To develop applets for web applications.
- To build GUIs and handle events generated by user interactions.
- To use the JDBC API to access database

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Java How to Program, 10th Edition, Paul Dietel, Harvey Dietel, Pearson Education.

Reference Books:

1. T. Budd “Understanding Object-Oriented Programming with Java”, updated edition, Pearson Education.
2. Cay S. Horstmann “Core Java Volume – 1 Fundamentals”,, Pearson Education.
3. Sagayaraj, Dennis, Karthik and Gajalakshmi “Java Programming for core and advanced learners, University Press
4. Y. Daniel Liang, “Introduction to Java programming”, Pearson Education.
5. P. Radha Krishna “Object Oriented Programming through Java”, University Press.
6. S. Malhotra, S. Chudhary, “Programming in Java”, 2nd edition, Oxford Univ. Press.
7. R.A. Johnson, “Java Programming and Object-oriented Application Development”, Cengage Learning.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
2	1	0	3

19A05402T DESIGN AND ANALYSIS OF ALGORITHMS

(Common to CSE & IT)

Course Objectives:

- To demonstrate the importance of algorithms in computing.
- To explain the analysis of algorithms
- To illustrate the method of finding the complexity of algorithms
- To explain the advanced algorithm design and analysis techniques.
- To introduce special classes of algorithms NP – completeness and the classes P and NP.

UNIT I

Introduction: Algorithm, Algorithm specification, Performance analysis.

Divide and Conquer: General method, Binary Search, Finding the maximum and minimum, Merge sort, Quick Sort, Selection, Strassen's matrix multiplication.

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

- Understand growth functions and Asymptotic notations
- Derive the recurrence equation for running time of a given algorithm and solve.
- Understand the general principle of Divide and Conquer and identify suitable problems to apply Divide and Conquer paradigm
- Analyze the time complexities of Binary Search, Finding the maximum and minimum, and Strassen's matrix multiplication algorithms.
- Compare complexities of Merge sort, Quick sort and Selection sort techniques

UNIT II

Greedy Method: General method, Knapsack problem, Job Scheduling with Deadlines, Minimum cost Spanning Trees, Optimal storage on tapes, Single-source shortest paths.

Dynamic programming: General Method, Multistage graphs, All-pairs shortest paths, Optimal binary search trees, 0/1 knapsack, the traveling salesperson problem.

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

- Understand optimization problems and the general principles of Greedy and Dynamic Programming paradigms to solve them.
- Apply subset and ordering paradigms of greedy strategy for Knapsack problem, Job Scheduling with Deadlines, Minimum cost Spanning Trees, Optimal storage on tapes, and finding Single-source shortest paths.
- Define Principle of optimality with examples.
- Differentiate Greedy and Dynamic programming paradigms.
- Apply dynamic programming strategy for Optimal binary search trees, Multistage graphs, All-pairs shortest paths, 0/1 knapsack, the traveling salesperson problem.

UNIT III

Basic Traversal and Search Techniques: Techniques for binary trees, Techniques for Graphs, Connected components and Spanning trees, Bi-connected components and DFS

Back tracking: General Method, 8 – queens problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles, Knapsack Problem.

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

- Define solution space tree.
- Illustrate graph search strategies : BFS, DFS and D-Search .
- Determine articulation points and bi-connected components in a given graph using Depth First Spanning Trees.
- Demonstrate the recursive and iterative backtracking algorithms.
- Apply backtracking strategy to solve N – queens problem, Sum of subsets problem and Knapsack problem.
- Apply backtracking to solve m-colorability optimization problem.
- Determine all possible Hamiltonian Cycles in a graph using backtracking algorithm.

UNIT IV

Branch and Bound: The method, Travelling salesperson, 0/1 Knapsack problem, Efficiency considerations.

Lower Bound Theory: Comparison trees, Lower bounds through reductions – Multiplying triangular matrices, inverting a lower triangular matrix, computing the transitive closure.

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

- Illustrate the state space search techniques; FIFO, LIFO and LC.
- Analyze the advantage of bounding functions in Branch and Bound technique to solve the Travelling Salesperson problem.
- Compare the LC and FIFO branch and bound solutions for 0/1 knapsack problem.
- Understand lower bound theory concept in solving algebraic problems.

UNIT V

NP – Hard and NP – Complete Problems: NP Hardness, NP Completeness, Consequences of being in P, Cook‘s Theorem, Reduction Source Problems, Reductions: Reductions for some known problems

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

- Differentiate deterministic and Non-deterministic algorithms.
- Define P, NP, NP –hard and NP-complete classes of problems.
- Understand the satisfiability problem.
- State Cook’s Theorem.
- Understand the reduction techniques.

Course outcomes

- Determine the time complexity of an algorithm by solving the corresponding recurrence equation
- Apply the Divide and Conquer strategy to solve searching, sorting and matrix multiplication problems.
- Analyze the efficiency of Greedy and Dynamic Programming design techniques to solve the optimization problems.
- Apply Backtracking technique for solving constraint satisfaction problems.
- Analyze the LC and FIFO branch and bound solutions for optimization problems, and compare the time complexities with Dynamic Programming techniques.
- Define and Classify deterministic and Non-deterministic algorithms; P, NP, NP –hard and NP-complete classes of problems.

Text Books

1. Ellis Horowitz, SartajSahni and Rajasekaran, “Fundamentals of Computer Algorithms”, 2nd Edition, 2012, University Press.
2. ParagHimanshu Dave and HimanshuBhalchandra Dave, “Design and Analysis of Algorithms”, Second Edition, Pearson Education.

References

1. AnanyLevitin, “Introduction to the Design and Analysis of Algorithms”, Third Edition, Pearson Education, 2012.
2. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.
3. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
4. Donald E. Knuth, “The Art of Computer Programming”, Volumes 1& 3 Pearson Education, 2009. Steven S. Skiena, “The Algorithm Design Manual”, Second Edition, Springer, 2008.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem **L T P C**
0 0 3 1.5

19A05302P DATABASE MANAGEMENT SYSTEMS LABORATORY

(Common to CSE& IT)

Course Objectives:

1. To implement the basic knowledge of SQL queries and relational algebra.
2. To construct database models for different database applications.
3. To apply normalization techniques for refining of databases.
4. To practice various triggers, procedures, and cursors using PL/SQL.
5. To design and implementation of a database for an organization

Week-1: CREATION OF TABLES

1. Create a table called Employee with the following structure.

Name	Type
Empno	Number
Ename	Varchar2(20)
Job	Varchar2(20)
Mgr	Number
Sal	Number

a. Add a column commission with domain to the Employee table.
b. Insert any five records into the table.
c. Update the column details of job
d. Rename the column of Employ table using alter command.
e. Delete the employee whose empno is19.

2. Create department table with the followings tructure.

Name	Type
Deptno	Number
Deptname	Varchar2(20)
location	Varchar2(20)

a. Add column designation to the departmenttable.
b. Insert values into thetable.
c. List the records of emp table grouped bydeptno.
d. Update the record where deptno is9.
e. Delete any column data from thetable

3. Create a table called Customer table

Name	Type
Cust name	Varchar2(20)
Cust street	Varchar2(20)
Cust city	Varchar2(20)

a. Insert records into thetable.
b. Add salary column to thetable.
c. Alter the table columndomain.
d. Drop salary column of the customertable.
e. Delete the rows of customer table whose ust_city is ‘hyd’.
f. Create a table called branchtable.

Name	Type
Branch name	Varchar2(20)
Branch city	Varchar2(20)
asserts	Number

4. Increase the size of data type for asserts to thebranch.
   a. Add and drop a column to the branchtable.
   b. Insert values to thetable.
   c. Update the branch namecolumn
   d. Delete any two columns from the table

5. Create a table called sailor table

Name	Type
Sid	Number
Sname	Varchar2(20)
rating	Varchar2(20)

   a. Add column age to the sailortable.
   b. Insert values into the sailortable.
   c. Delete the row with rating>8.
   d. Update the column details ofsailor.
   e. Insert null values into thetable.

6. Create a table called reservestable

Name	Type
Boat id	Integer
sid	Integer
day	Integer

   a. Insert values into the reservestable.
   b. Add column time to the reservestable.
   c. Alter the column day data type todate.
   d. Drop the column time in thetable.`
   e. Delete the row of the table with somecondition.

Week-2: QUERIES USING DDL AND DML

1. a. Create a user and grant all permissions to theuser.
   b. Insert the any three records in the employee table and use rollback. Check theresult.
   c. Add primary key constraint and not null constraint to the employeetable.
   d. Insert null values to the employee table and verify theresult.
2. a. Create a user and grant all permissions to theuser.
   b. Insert values in the department table and usecommit.
   c. Add constraints like unique and not null to the departmenttable.
   d. Insert repeated values and null values into thetable.
3. a. Create a user and grant all permissions to theuser.
   b. Insert values into the table and usecommit.
   c. Delete any three records in the department table and userollback.
   d. Add constraint primary key and foreign key to thetable.

4. a. Create a user and grant all permissions to theuser.
   b. Insert records in the sailor table and usecommit.
   c. Add save point after insertion of records and verify savepoint.
   d. Add constraints not null and primary key to the sailortable.
5. a. Create a user and grant all permissions to theuser.
   b. Use revoke command to remove userpermissions.
   c. Change password of the usercreated.
   d. Add constraint foreign key and notnull.
6. a. Create a user and grant all permissions to theuser.
   b. Update the table reserves and use savepoint androllback.
   c. Add constraint primary key , foreign key and not null to the reservestable
   d. Delete constraint not null to the tablecolumn

Week-3:QUERIES USING AGGREGATE FUNCTIONS

1. a. By using the group by clause, display the enames who belongs to deptno 10 alongwith averagesalary.
   b. Display lowest paid employee details under eachdepartment.
   c. Display number of employees working in each department and their departmentnumber.
   d. Using built in functions, display number of employees working in each department and their department name from dept table. Insert deptname to dept table and insert dept name for each row, do the required thing specified above.
   e. List all employees which start with either B or C.
   f. Display only these ename of employees where the maximum salary is greater than or equalto 5000.
2. a. Calculate the average salary for each differentjob.
   b. Show the average salary of each job excludingmanager.
   c. Show the average salary for all departments employing more than threepeople.
   d. Display employees who earn more than thelowest salary in department 30
   e. Show that value returned by sign (n)function.
   f. How many days between day of birth to currentdate
3. a. Show that two substring as singlestring.
   b. List all employee names, salary and 15% rise insalary.
   c. Display lowest paid emp details under eachmanager
   d. Display the average monthly salary bill for eachdeptno.
   e. Show the average salary for all departments employing more than twopeople.

f. By using the group by clause, display the eid who belongs to deptno 05 along withaverage salary.

4. a. Count the number of employees in department20
   b. Find the minimum salary earned byclerk.
   c. Find minimum, maximum, average salary of allemployees.
   d. List the minimum and maximum salaries for each jobtype.
   e. List the employee names in descendingorder.
   f. List the employee id, names in ascending order byempid.
5. a. Find the sids ,names of sailors who have reserved all boats called“INTERLAKE
   Find the age of youngest sailor who is eligible to vote for each rating level with at least two such sailors.
   b. Find the sname , bid and reservation date for eachreservation.
   c. Find the ages of sailors whose name begin and end with B and has at least 3characters.
   d. List in alphabetic order all sailors who have reserved redboat.
   e. Find the age of youngest sailor for each ratinglevel.
6. a. List the Vendors who have delivered products within 6 months from orderdate.
   b. Display the Vendor details who have supplied both Assembled and Subparts.
   c. Display the Sub parts by grouping the Vendor type (Local or NonLocal).
   d. Display the Vendor details in ascendingorder.
   e. Display the Sub part which costs more than any of the Assembledparts.
   f. Display the second maximum cost Assembledpart

Week-4: PROGRAMS ON PL/SQL

1. a. Write a PL/SQL program to swaptwonumbers.
   b. Write a PL/SQL program to find the largest of threenumbers.
2. a. Write a PL/SQL program to find the total and average of 6 subjects and display thegrade.
   b. Write a PL/SQL program to find the sum of digits in a givennumber.
3. a. Write a PL/SQL program to display the number in reverseorder.
   b. Writea PL/SQLprogramto checkwhetherthegiven numberisprimeornot.
4. a. Write a PL/SQL program to find the factorial of a givennumber.
   b. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius andarea.
5. a. Write a PL/SQL program to accept a string and remove the vowels from the string. (When ‘hello’ passed to the program it should display ‘Hll’ removing e and o from the

worldHello).

b. Write a PL/SQL program to accept a number and a divisor. Make sure the divisor is less than or equal to 10. Else display an error message. Otherwise Display the remainderin words.

Week-5: PROCEDURES AND FUNCTIONS

1. Write a function to accept employee number as parameter and return Basic +HRA together as single column.
2. Accept year as parameter and write a Function to return the total net salary spent for a givenyear.
3. Create a function to find the factorial of a given number and hence findNCR.
4. Write a PL/SQL block o pint prime Fibonacci series using localfunctions.
5. Create a procedure to find the lucky number of a given birthdate.
6. Create function to the reverse of given number

Week-6: TRIGGERS

1. Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and newvalues:

   CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Alive	24	Khammam	2000
2	Bob	27	Kadappa	3000
3	Catri	25	Guntur	4000
4	Dena	28	Hyderabad	5000
5	Eeshwar	27	Kurnool	6000
6	Farooq	28	Nellur	7000

2. Creation of insert trigger, delete trigger, update trigger practice triggers using the passenger database.
   Passenger(Passport_ id INTEGER PRIMARY KEY, Name VARCHAR (50) NotNULL, Age Integer Not NULL, Sex Char, Address VARCHAR (50) NotNULL);
   a. Write a Insert Trigger to check the Passport_id is exactly six digits ornot.
   b. Write a trigger on passenger to display messages ‘1 Record is inserted’, ‘1 record is deleted’, ‘1 record is updated’ when insertion, deletion and updation are done on passenger respectively.

3. Insert row in employee table using Triggers. Every trigger is created with name any trigger have same name must be replaced by new name. These triggers can raised before insert, update or delete rows on data base. The main difference between a trigger and a stored procedure is that the former is attached to a table and is only fired when an INSERT, UPDATE or DELETEoccurs.
4. Convert employee name into uppercase whenever an employee record is inserted or updated. Trigger to fire before the insert orupdate.
5. Trigger before deleting a record from emp table. Trigger will insert the row to be deleted into table called delete _emp and also record user who has deleted the record and date and time ofdelete.
6. Create a transparent audit system for a table CUST_MSTR. The system must keep track of the records that are being deleted orupdated

Week-7:PROCEDURES

1. Create the procedure for palindrome of given number.
2. Create the procedure for GCD: Program should load two registers with two Numbers and then apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number by the smaller number till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisors of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.
3. Write the PL/SQL programs to create the procedure for factorial of given number.
4. Write the PL/SQL programs to create the procedure to find sum of N natural number.
5. Write the PL/SQL programs to create the procedure to find Fibonacci series.
6. Write the PL/SQL programs to create the procedure to check the given number is perfect or not

Week-8: CURSORS

1. Write a PL/SQL block that will display the name, dept no, salary of fist highest paid employees**.**
2. Update the balance stock in the item master table each time a transaction takes place in the item transaction table. The change in item master table depends on the item id is already present in the item master then update operation is performed to decrease the balance stock by the quantity specified in the item transaction in case the item id is not present in the item master table then the record is inserted in the item master table.
3. Write a PL/SQL block that will display the employee details along with salary using cursors.

4. To write a Cursor to display the list of employees who are working as a Managers or Analyst.
5. To write a Cursor to find employee with given job and deptno.
6. Write a PL/SQL block using implicit cursor that will display message, the salaries of all the employees in the ‘employee’ table are updated. If none of the employee’s salary are updated we geta message 'None of the salaries were updated'. Else we get a message like for example, 'Salaries for 1000 employees are updated' if there are 1000 rows in ‘employee’ table

Week-9: 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:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-10: **CASE STUDY GENERAL HOSPITAL**

AGeneralHospitalconsistsofanumberofspecializedwards(suchasMaternity,Pediatric,Oncology, etc). Each ward hosts a number of patients, who were admitted on the recommendation of their ownGP 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, do the following.

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-11: **CASE STUDY: CAR RENTAL COMPANY**

A database is to be designed for a car rental company. The information required includes a description of cars, subcontractors (i.e. garages), company expenditures, company revenues and customers. Cars are to be described by such data as: make, model, year of production, engine size, fuel type, number of passengers, registration number, purchase price, purchase date, rent price and insurance details. It is the company policy not to keep any car for a period exceeding one year. All major repairs and maintenance are done by subcontractors (i.e. franchised garages), with whom CRC has long-term agreements. Therefore the data about garages to be kept in the database includes garage names, addresses, range of services and the like. Some garages require payments immediately after a repair has been made; with others CRC has made arrangements for credit facilities. Company expenditures are to be registered for all outgoings connected with purchases, repairs, maintenance, insurance etc. Similarly the cash inflow coming from all sources: Car hire, car sales, insurance claims must be kept of file. CRC maintains a reasonably stable client base. For this privileged category of customers special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual customers must pay a deposit for an estimated time of rental, unless they wish to pay by credit card. All major credit cards are accepted. Personal details such as name, address, telephone number, driving license, number about each customer are kept in the database. For the above case study, do the following:

1. Analyze the data required.
2. Normalize the attributes.

Create the logical data model using E-R diagrams

Week-12: **CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM**

A database is to be designed for a college to monitor students' progress throughout their course of study. The students are reading for a degree (such as BA, BA (Hons) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from. A module is coordinated by a module leader who shares teaching duties with one or more lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre- requisites modules and some degree programmes have compulsory modules. The database is also to contain some information about

studentsincludingtheirnumbers,names,addresses,degreestheyreadfor,andtheirpastperformanc
e

i.e. modules taken and examination results. For the above case study, do the following:

1. Analyze the datarequired.
2. Normalize theattributes.
3. Create the logical data model i.e., ERdiagrams.
4. Comprehend the data given in the case study by creating respective tables with primary keys and foreign keys whereverrequired.
5. Insert values into the tables created (Be vigilant about Master- Slavetables).
6. Display the Students who have taken M.Sccourse
7. Display the Module code and Number of Modules taught by eachLecturer.
8. Retrieve the Lecturer names who are not Module Leaders.
9. Display the Department name which offers ‘English’module.
10. Retrieve the Prerequisite Courses offered by every Department (with Departmentnames).
11. Present the Lecturer ID and Name who teaches‘Mathematics’.
12. Discover the number of years a Module istaught.
13. List out all the Faculties who work for ‘Statistics’Department.
14. List out the number of Modules taught by each ModuleLeader.
15. List out the number of Modules taught by a particularLecturer.
16. Create a view which contains the fields of both Department and Module tables. (Hint- The fields like Module code, title, credit, Department code and itsname).
17. Update the credits of all the prerequisite courses to 5. Delete the Module ‘History’ from the Moduletable.

Course outcomes:

Students should be able to

1. Design database for any real world problem
2. Implement PL/SQL programs
3. Define SQL queries
4. Decide the constraints
5. Investigate for data inconsistency

Reference Books:

1. RamezElmasri, Shamkant, B. Navathe, "Database Systems", Pearson Education, 6th Edition, 2013.
2. Peter Rob, Carles Coronel, "Database System Concepts", Cengage Learning, 7th Edition, 2008.

Web References:

http://www.scoopworld.in

SOFTWARE AND HARDWARE REQUIREMENTS FOR A BATCH OF 24 STUDENTS:

HARDWARE: Desktop Computer Systems: 24 nos

SOFTWARE: Oracle 11g.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
0	0	3	1.5

19A05303P OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

(Common to CSE & IT)

Course Objectives

- To introduce the concepts of Java.
- To Practice object-oriented programs and build java applications.
- To implement java programs for establishing interfaces.
- To implement sample programs for developing reusable software components.
- To establish database connectivity in java and implement GUI applications.

Week-1

a. Installation of Java software, study of any Integrated development environment, Use Eclipse or Netbean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with java program to find prime numbers between 1 to n.

b. Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.

c. Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection

(i.e domestic or commercial). Commute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- First 100 units - Rs. 1 per unit
- 101-200 units - Rs. 2.50 per unit
- 201 -500 units - Rs. 4 per unit
- > 501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- First 100 units - Rs. 2 per unit
- 101-200 units - Rs. 4.50 per unit
- 201 -500 units - Rs. 6 per unit
- > 501 units - Rs. 7 per unit

d. Write a Java program to multiply two given matrices.

Week-2

a. Write Java program on use of inheritance, preventing inheritance using final, abstract classes.

b. Write Java program on dynamic binding, differentiating method overloading and overriding.

c. Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen) using Interfaces.

Week-3

a. Write Java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.

b. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

c. Write a Java program to read the time intervals (HH:MM) and to compare system time if the system Time between your time intervals print correct time and exit else try again to repute the same thing. By using StringToknizer class.

Week-4

a. Write a Java program to implement user defined exception handling.

b. Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read. Display the complete set of unique values input after the user enters each new value.

Week-5

a. Write a Java program that creates a user interface to perform integer division. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

b. Write a Java program that creates three threads. First thread displays ―Good Morning‖ every one second, the second thread displays ―Hello‖ every two seconds and the third thread displays ―Welcome‖ every three seconds.

Week-6

a. Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part where n is the sequence number of the part file.

b. Write a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

Week-7

a. Write a java program that displays the number of characters, lines and words in a text file.
b. Write a java program that reads a file and displays the file on the screen with line number before each line.

Week-8

a. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
b. Develop a Java application for stack operation using Buttons and JOptionPane input and Message dialog box.
c. Develop a Java application to perform Addition, Division, Multiplication and substraction using JOptionPane dialog Box and Textfields.

Week-9

a. Develop a Java application for the blinking eyes and mouth should open while blinking.
b. Develop a Java application that simulates a traffic light. The program lets the user select one of three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with ―STOP‖ or ―READY‖ or ‖GO‖ should appear above the buttons in selected color. Initially, there is no message shown.

Week-10

a. Develop a Java application to implement the opening of a door while opening man should present before hut and closing man should disappear.
b. Develop a Java application by using JtextField to read decimal value and converting a decimal number into binary number then print the binary value in another JtextField.

Week-11

a. Develop a Java application that handles all mouse events and shows the event name at the center of the window when a mouse event is fired. Use adapter classes.
b. Develop a Java application to demonstrate the key event handlers.

Week-12

a. Develop a Java application to find the maximum value from the given type of elements using a generic function.
b. Develop a Java application that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result.
c . Develop a Java application for handling mouse events.

Week-13

a. Develop a Java application to establish a JDBC connection, create a table student with properties name, register number, mark1,mark2, mark3. Insert the values into the table by using the java and display the information of the students at front end.

Course Outcomes:

On successful completion of this laboratory students will be able to:

1. Recognize the Java programming environment.
2. Develop efficient programs using multithreading.
3. Design reliable programs using Java exception handling features.
4. Extend the programming functionality supported by Java.
5. Select appropriate programming construct to solve a problem.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
0	0	3	1.5

19A05402P DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

LIST OF EXPERIMENTS

Course objectives

1. Learn how to analyze a problem and design the solution for the problem.
2. Design and implement efficient python programming for a specified application.
3. Identify and apply the suitable algorithm for the given real world problem.

Week-l QUICK SORT

Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Week-2 MERGE SORT

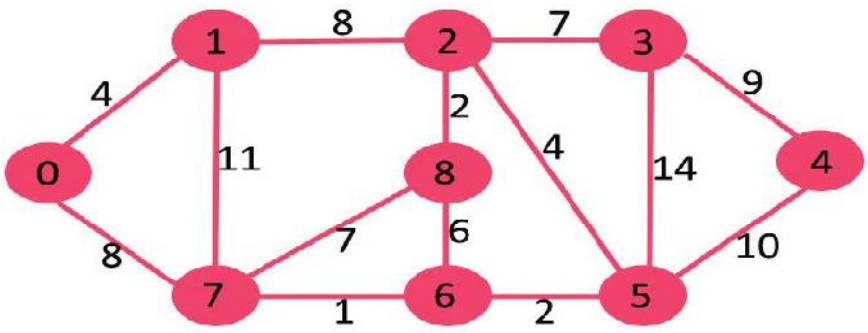
Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

Week-3 KNAPSACK PROBLEM

Implement 0/1 Knapsack problem using Dynamic Programming.

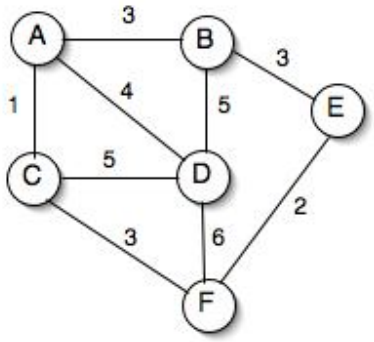
Week-4 SHORTEST PATHS ALGORITHM

From a given vertex in a weighted connected graph, find shortest paths from 0 to other vertices using Dijkstra‘s algorithm.



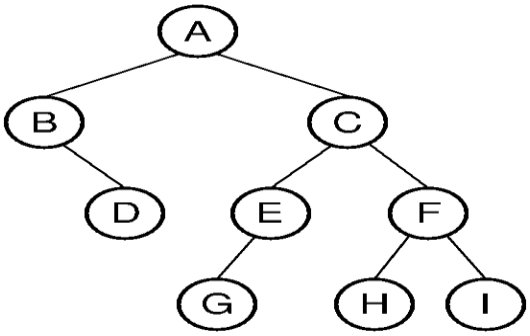
Week-5 MINIMUM COST SPANNING TREE

Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal‘s algorithm.

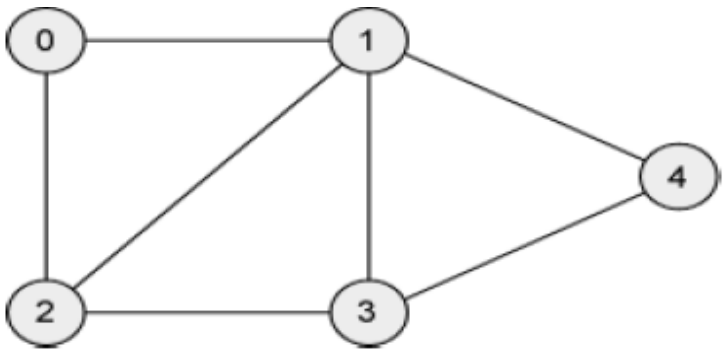


Week-6 TREE TRAVESRSALS

Perform various tree traversal algorithms for a given tree.

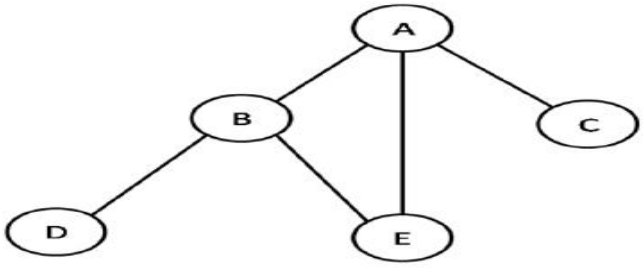


Week-7 GRAPH TRAVERSALS



a. Print all the nodes reachable from a given starting node in a digraph using BFS method.

b. Check whether a given graph is connected or not using DFS method.



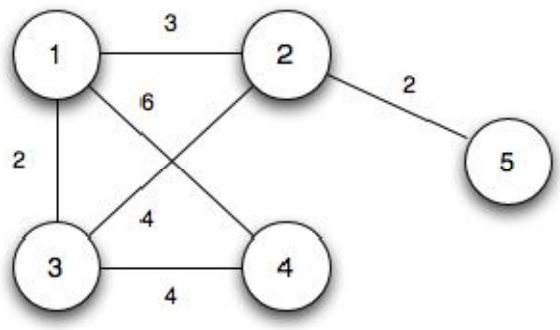
Week-8 SUM OF SUB SETS PROBLEM

Find a subset of a given set S = {sl, s2,....., sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2, 6} and {1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.

Week-9 TRAVELLING SALES PERSON PROBLEM

Implement any scheme to find the optimal solution for the Traveling Sales Person problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation

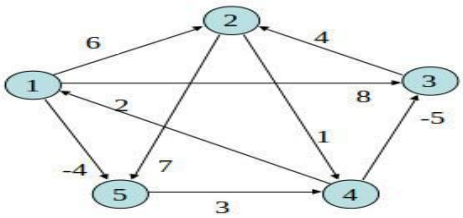
Week-10 MINIMUM COST SPANNING TREE



Find Minimum Cost Spanning Tree of a given undirected graph using Prim‘s algorithm.

Week-11 ALL PAIRS SHORTEST PATHS

Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.



	1	2	3	4	5
1	0	6	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	3	0

Week-12 N QUEENS PROBLEM

Implement N Queen's problem using Back Tracking.

Reference Books:

1. Levitin A, ―Introduction to the Design and Analysis of Algorithms‖, Pearson Education, 2008.
1. Goodrich, M.T. R Tomassia, ―Algorithm Design foundations Analysis and Internet Examples‖, John Wiley and Sons, 2006.
2. Base Sara, Allen Van Gelder, ―Computer Algorithms Introduction to Design and Analysis‖, Pearson, 3rd
3. Edition, 1999.

Web Reference:

1. http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html
2. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
3. http://www.facweb.iitkgp.ernet.in/~sourav/daa.html

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
3	0	0	0

19A99302 BIOLOGY FOR ENGINEERS

Course Objectives: To provide basic understanding about life and life Process. Animal an plant systems. To understand what bimolecules, are, their structures are functions. Application of certain bimolecules in Industry.

- Brief introduction about human physiology and bioengineering.
- To understand hereditary units, i.e. DNA (genes) and RNA and their synthesis in living organism.
- How biology Principles can be applied in our daily life using different technologies.
- Brief introduction to the production of transgenic microbes, Plants and animals.

Unit I: Introduction to Basic Biology

Cell as Basic unit of life, cell theory, Cell shapes, Cell structure, Cell cycle. Chromosomes. Prokaryotic and eukaryotic Cell. Plant Cell, Animal Cell, Plant tissues and Animal tissues, Brief introduction to five kingdoms of classification.

Unit Outcomes:

After completing this unit, the student will be able to

- Summarize the basis of life. (L1)
- Understand the difference between lower organisms (prokaryotes) from higher organisms (eukaryotes). (L2)
- Understand how organisms are classified. (L3)

Unit II: Introduction to Biomolecules

Carbohydrates, lipids, proteins, Vitamins and minerals, Nucleic acids (DNA and RNA) and their types. Enzymes, Enzyme application in Industry. Large scale production of enzymes by Fermentation.

Unit Outcomes:

After completing this unit, the student will be able to

- Understand what are biomolecules? their role in living cells, their structure, function and how they are produced. (L1)
- Interpret the relationship between the structure and function of nucleic acids. (L2)
- Summarize the applications of enzymes in industry. (L3)
- Understand what is fermentation and its applications of fermentation in industry. (L4)

Unit III: Human Physiology

Nutrition: Nutrients or food substances. Digestive system, Respiratory system, (aerobic and anaerobic Respiration). Respiratory organs, respiratory cycle. Excretory system.

Unit Outcomes:

After completing this unit, the student will be able to

- Understand what nutrients are (L1)
- Understand the mechanism and process of important human functions (L2 & L3)

Unit IV: Introduction to Molecular Biology and recombinant DNA Technology

Prokaryotic gene and Eukaryotic gene structure. DNA replication, Transcription and Translation. rDNA technology. Introduction to gene cloning.

Unit Outcomes:

After completing this unit, the student will be able to

- Understand and explain about gene structure and replication in prokaryotes and Eukaryotes (L1)
- How genetic material is replicated and also understands how RNA and proteins are synthesized. (L2)
- Understand about recombinant DNA technology and its application in different fields.(L3)
- Explain what is cloning. (L4)

Unit V: Application of Biology

Brief introduction to industrial Production of Enzymes, Pharmaceutical and therapeutic Proteins, Vaccines and antibodies. Basics of biosensors, biochips, Bio fuels, and Bio Engineering. Basics of Production of Transgenic plants and animals.

Unit Outcomes:

After completing this unit, the student will be able to Understand.

- How biology is applied for production of useful products for mankind.(L1)
- What are biosensors, biochips etc. (L2)
- Understand transgenic plants and animals and their production (L3)

Course Outcomes:

After studying the course, the student will be able to:

- Explain about cells and their structure and function. Different types of cells and basics for classification of living Organisms.
- Explain about biomolecules, their structure and function and their role in the living organisms. How biomolecules are useful in Industry.
- Briefly about human physiology.
- Explain about genetic material, DNA, genes and RNA how they replicate, pass and preserve vital information in living Organisms.
- Know about application of biological Principles in different technologies for the production of medicines and Pharmaceutical molecules through transgenic microbes, plants and animals.

Text books:

1. P.K.Gupta, Cell and Molecular Biology, 5th Edition, Rastogi Publications -
2. U. Satyanarayana. Biotechnology, Books & Allied Ltd 2017

Reference Books:

1. N. A. Campbell, J. B. Reece, L. Urry, M. L. Cain and S. A. Wasserman, "Biology: A Global Approach", Pearson Education Ltd, 2018.
2. T Johnson, Biology for Engineers, CRC press, 2011
3. J.M. Walker and E.B. Gingold, Molecular Biology and Biotechnology 2nd ed.. Panima Publications. PP 434.
4. David Hames, Instant Notes in Biochemistry –2016
5. Phil Tunner, A. Mctennan, A. Bates & M. White, Instant Notes – Molecular Biology –-2014

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
3	0	0	3

19A54401 NUMBER THEORY AND APPLICATIONS

(Common to CSE & IT)

Course Objective:

This course enables the students to learn the concepts of number theory and its applications to information security.

Unit-I-Integers, Greatest common divisors and prime Factorization

The well-ordering property-Divisibility-Representation of integers-Computer operations with integers-Prime numbers-Greatest common divisors-The Euclidean algorithm -The fundamental theorem of arithmetic-Factorization of integers and the Fermat numbers-Linear Diophantine equations

Unit Outcomes:

Students will be able to

1. Understand basics of number theory concepts.
2. Solve problems on prime numbers.
3. Understand Euclidean algorithm and its applications.

Unit-II-Congruences

Introduction to congruences -Linear congruences-The Chinese remainder theorem-Systems of linear congruences

Unit Outcomes:

Students will be able to

1. understand Congruences and its basic properties.
2. understand Chinese remainder theorem and its applications.

Unit-III Applications of Congruences

Divisibility tests-The perpetual calendar-Round-robin tournaments-Computer file storage and hashing functions. Wilson's theorem and Fermat's little theorem- Pseudo primes- Euler's theorem- Euler's p hi-function- The sum and number of divisors- Perfect numbers and Mersenne primes.

Unit Outcomes:

Students will be able to

1. Understand divisibility tests.
2. Apply the concept of congruences to various applications.
3. Understand various theorems on Number theory and its applications.

Unit-IV- Finite fields & Primality, factoring

Finite fields- quadratic residues and reciprocity-Pseudo primes-rho method-fermat factorization and factor bases.

Unit Outcomes:

Students will be able to

1. Understand the terminology of finite fields.
2. Understand rho method and fermat factorization.

Unit-V- Cryptology

Basic terminology-complexity theorem-Character ciphers-Block ciphers-Exponentiation ciphers-Public-key cryptography-Discrete logarithm-Knapsack ciphers- RSA algorithm-Some applications to computer science.

Unit Outcomes:

Students will be able to

1. Understand the terminology of cryptology.
2. Understand different encryption mechanisms.

Text Books:

1. Elementary number theory and its applications, Kenneth H Rosen, AT & T Information systems & Bell laboratories.
2. A course in Number theory & Cryptography, Neal Koblitz, Springer.

Reference Books:

1. An Introduction To The Theory Of Numbers, Herbert S. Zuckerman, Hugh L. Montgomery, Ivan Niven, wiley publishers
2. Introduction to Analytic number theory-Tom M Apostol, springer
3. Elementary number theory, VK Krishnan, Universities press

Course Outcomes:

After the completion of course, student will be able to

- Understand number theory and its properties.
- Understand principles on congruences
- Develop the knowledge to apply various applications
- Develop various encryption methods and its applications.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
3	0	0	3

19A05401 COMPUTER ORGANIZATION (CSE & IT)

Course Objectives:

- To learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design
- To understand the structure and behavior of various functional modules of a computer.
- To learn the techniques that computers use to communicate with I/O devices
- To acquire the concept of pipelining and exploitation of processing speed.
- To learn the basic characteristics of multiprocessors

UNIT - I

Basic Structure of Computer: Computer Types, Functional Units, Basic operational Concepts, Bus Structure, Software, Performance, Multiprocessors and Multicomputer.
Machine Instructions and Programs: Numbers, Arithmetic Operations and Programs, Instructions and Instruction Sequencing, Addressing Modes, Basic Input/output Operations, Stacks and Queues, Subroutines, Additional Instructions.

Unit Outcomes:
Student is able to

- Identify the basic functional units and different ways of interconnecting to form a computer system.
- Illustrate various addressing modes for accessing register and memory operands.
- Describe the instruction sequencing and various types of instructions.

UNIT - II

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations.
Basic Processing Unit: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Multi programmed Control.

Unit Outcomes:
Student is able to

- Outline the arithmetic operations on signed numbers.
- Describe the operations performed on floating point numbers.
- Distinguish between hardwired and micro programmed control units.

UNIT - III

The Memory System: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

Unit Outcomes:
Student is able to

- Recognize the various types of memories.
- Analyze the performance of cache memory.
- Apply effective memory management strategies.

UNIT - IV

Input/Output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

Unit Outcomes:
Student is able to

- Examine the basics of I/O data transfer synchronization.
- Analyze the interrupt handling mechanisms of various processors.
- Describe various techniques for I/O data transfer methods.

UNIT - V

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets.

Large Computer Systems: Forms of Parallel Processing, Array Processors, The Structure of General-Purpose multiprocessors, Interconnection Networks.

Unit Outcomes:

Student is able to

- Investigate the use of pipelining and multiple functional units in the design of high-performance processors.
- Design and analyze a high performance processor.
- Describe the interconnection networks for multiprocessors.

Course Outcomes:

At end of the course the student will be able to

- Understand computer architecture concepts related to design of modern processors, memories and I/Os
- Identify the hardware requirements for cache memory and virtual memory
- Design algorithms to exploit pipelining and multiprocessors
- Understand the importance and tradeoffs of different types of memories.
- Identify pipeline hazards and possible solutions to those hazards

TEXT BOOKS:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, McGraw Hill Education, 2013.

REFERENCE BOOKS:

1. M.Morris Mano, "Computer System Architecture", 3rd Edition, Pearson Education.
2. Themes and Variations, Alan Clements, "Computer Organization and Architecture", CENGAGE Learning.
3. Smruti Ranjan Sarangi, "Computer Organization and Architecture", McGraw Hill Education.
4. John P.Hayes, "Computer Architecture and Organization", McGraw Hill Education

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
3	0	0	3

19A12401 FORMAL LANGUAGES AND AUTOMATA THEORY

Course Objectives:

This course is designed to:

1. Introduce languages, grammars, and computational models
2. Explain the Context Free Grammars
3. Enable the students to use Turing machines
4. Demonstrate decidability and un-decidability for NP Hard problems

UNIT – I: Finite Automata

Why Study Automata Theory? The Central Concepts of Automata Theory, Automation, Finite Automation, Transition Systems, Acceptance of a String by a Finite Automation, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with E-Transition, Minimization of Finite Automata, Mealy and Moore Machines, Applications and Limitation of Finite Automata.

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

- ➢ Distinguish DFA and NFA.
- ➢ Construct DFA for an input string.
- ➢ Perform minimization of Automata.
- ➢ Compare Moore and Mealy Machines.

UNIT – II: Regular Expressions

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two Regular Expressions, Manipulations of Regular Expressions, Finite Automata, and Regular Expressions, Inter Conversion, Equivalence between Finite Automata and Regular Expressions, Pumping Lemma, Closers Properties, Applications of Regular Expressions, Finite Automata and Regular Grammars, Regular Expressions and Regular Grammars.

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

- ➢ Construct regular expression for the given Finite Automata.
- ➢ Construct finite automata for the given regular expression.
- ➢ Apply closure properties on regular expressions.

UNIT – III: Context Free Grammars

Formal Languages, Grammars, Classification of Grammars, Chomsky Hierarchy Theorem, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars-Elimination of Useless Symbols, E-Productions and Unit Productions, Normal Forms for Context Free Grammars-Chomsky Normal Form and Greibach Normal Form, Pumping Lemma, Closure Properties, Applications of Context Free Grammars.

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

- ➢ Define Context Free Grammar.
- ➢ Distinguish Chomsky Normal Form and Greibach Normal form.
- ➢ Apply Pumping Lemma theorem on Context Free Grammar.

UNIT – IV: Pushdown Automata

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description Language Acceptance of pushdown Automata, Design of Pushdown Automata, Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars Conversion, Two Stack Pushdown Automata, Application of Pushdown Automata.

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

- ➢ List the applications of Pushdown Automata.
- ➢ Construct Pushdown Automata for context free grammar.

UNIT – V: Turing Machine

Turing Machine, Definition, Model, Representation of Turing Machines-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language of a Turing Machine, Design of Turing Machines, Techniques for Turing Machine Construction, Types of Turing Machines, Church's Thesis, Universal Turing Machine, Restricted Turing Machine.

Decidable and Undecidable Problems: NP, NP-Hard and NP-Complete Problems.

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

- ➢ List types of Turing Machines.
- ➢ Design Turing Machine.
- ➢ Formulate decidability and undecidability problems.

Course Outcomes:

Students will be able to:

- Explain formal machines, languages and computations (L2)
- Design finite state machines for acceptance of strings (L6)
- Develop context free grammars for formal languages (L3)
- Build pushdown automata for context free grammars (L3)
- Apply Turing machine for solving problems (L3)
- Validate decidability and undecidability (L6)

TEXT BOOKS:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. Theory of Computer Science-Automata, Languages and Computation, K.L.P. Mishra and N.Chandrasekaran, 3rd Edition, PHI, 2007.

REFERENCE BOOKS:

1. Formal Language and Automata Theory, K.V.N. Sunitha and N.Kalyani, Pearson, 2015.
2. Introduction to Automata Theory, Formal Languages and Computation, Shyamalendu Kandar, Pearson, 2013.
3. Theory of Computation, V. Kulkarni, Oxford University Press, 2013.
4. Theory of Automata, Languages and Computation, Rajendra Kumar, McGraw Hill, 2014.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
3	0	0	3

19A05404T SOFTWARE ENGINEERING

(Common to CSE & IT)

Course Objectives:

1. To learn the basic concepts of software engineering and life cycle models
2. To explore the issues in software requirements specification and enable to write SRS documents for software development problems
3. To elucidate the basic concepts of software design and enable to carry out procedural and object oriented design of software development problems
4. To understand the basic concepts of black box and white box software testing and enable to design test cases for unit, integration, and system testing
5. To reveal the basic concepts in software project management

Unit – I: Basic concepts in software engineering and software project management

Basic concepts: abstraction versus decomposition, evolution of software engineering techniques, Software development life cycle (SDLC) models: Iterative waterfall model, Prototype model, Evolutionary model, Spiral model, RAD model, Agile models, software project management: project planning, project estimation, COCOMO, Halstead's Software Science, project scheduling, staffing, Organization and team structure, risk management, configuration management.

Unit Outcomes:
Student should be able to

1. Recognize the basic issues in commercial software development.
2. Summarize software lifecycle models.
3. Infer Workout project cost estimates using COCOMO and schedules using PERT and GANTT charts.

Unit – II: Requirements analysis and specification

The nature of software, The Unique nature of Webapps, Software Myths, Requirements gathering and analysis, software requirements specification, Traceability, Characteristics of a Good SRS Document, IEEE 830 guidelines, representing complex requirements using decision tables and decision trees, overview of formal system development techniques. axiomatic specification, algebraic specification.

Unit Outcomes:

Student should be able to

- Identify basic issues in software requirements analysis and specification.
- Develop SRS document for sample problems using IEEE 830 format.
- Develop algebraic and axiomatic specifications for simple problems.

Unit – III : Software Design

Good Software Design, Cohesion and coupling, Control Hierarchy: Layering, Control Abstraction, Depth and width, Fan-out, Fan-in, Software design approaches, object oriented vs. function oriented design. Overview of SA/SD methodology, structured analysis, Data flow diagram, Extending DFD technique to real life systems, Basic Object oriented concepts, UML Diagrams, Structured design, Detailed design, Design review, Characteristics of a good user interface, User Guidance and Online Help, Mode-based Vs Mode-less Interface, Types of user interfaces, Component-based GUI development, User interface design methodology: GUI design methodology.

Unit Outcomes

Student should be able to

1. Identify the basic issues in software design.
2. Apply the structured, object oriented analysis and design (SA/SD) technique.
3. Recognize the basic issues in user interface design.

Unit – IV : Coding and Testing

Coding standards and guidelines, code review, software documentation, Testing, Black Box Testing, White Box Testing, debugging, integration testing, Program Analysis Tools, system testing, performance testing, regression testing, Testing Object Oriented Programs.

Unit Outcomes:

Student should be able to

- Identify the basic issues in coding practice.
- Recognize the basic issues in software testing.
- Design test cases for black box and white box testing.

Unit – V: Software quality, reliability, and other issues

Software reliability, Statistical testing, Software quality and management, ISO 9000, SEI capability maturity model (CMM), Personal software process (PSP), Six sigma, Software quality metrics, CASE and its scope, CASE environment, CASE support in software life cycle, Characteristics of software maintenance, Software reverse engineering, Software maintenance processes model, Estimation maintenance cost. Basic issues in any reuse program, Reuse approach, Reuse at organization level.

Unit Outcomes:

Student should be able to

- Summarize various methods of software quality management.
- Instruct the quality management standards ISO 9001, SEI CMM, PSP, and Six Sigma.
- Outline software quality assurance, quality measures, and quality control.
- Identify the basic issues in software maintenance, CASE support, and software reuse.

Course outcomes:

Student should be able to

- Obtain basic software life cycle activity skills.
- Design software requirements specification for given problems.
- Implement structure, object oriented analysis and design for given problems.
- Design test cases for given problems.
- Apply quality management concepts at the application level.

Text Book:

1. Rajib Mall, “Fundamentals of Software Engineering”, 5th Edition, PHI, 2018.
2. Pressman R, “Software Engineering- Practioner Approach”, McGraw Hill.

Reference Books:

1. Somerville, “Software Engineering”, Pearson 2.
2. Richard Fairley, “Software Engineering Concepts”, Tata McGraw Hill.
3. Jalote Pankaj, “An integrated approach to Software Engineering”, Narosa

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
2	1	0	3

19A05304T PYTHON PROGRAMMING

Course Objectives:

1. To learn the fundamentals of Python
2. To elucidate problem-solving using a Python programming language
3. To introduce a function-oriented programming paradigm through python
4. To get training in the development of solutions using modular concepts
5. To introduce the programming constructs of python

Unit – I

Introduction: What is a program, Running python, Arithmetic operators, Value and Types.

Variables, Assignments and Statements: Assignment statements, Script mode, Order of operations, string operations, comments.

Functions: Function calls, Math functions, Composition, Adding new Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters are local, Stack diagrams, Fruitful Functions and Void Functions, Why Functions.

Unit Outcomes:
Student should be able to

1. List the basic constructs of Python.
2. Solve the problems by applying modularity principle.

Unit – II

Case study: The turtle module, Simple Repetition, Encapsulation, Generalization, Interface design, Refactoring, docstring.

Conditionals and Recursion: floor division and modulus, Boolean expressions, Logical operators, Conditional execution, Alternative execution, Chained conditionals, Nested conditionals, Recursion, Infinite Recursion, Keyboard input.

Fruitful Functions: Return values, Incremental development, Composition, Boolean functions, More recursion, Leap of Faith, Checking types,

Unit Outcomes:
Student should be able to

1. Apply the conditional execution of the program.
2. Apply the principle of recursion to solve the problems.

Unit - III

Iteration: Reassignment, Updating variables, The while statement, Break, Square roots, Algorithms.

Strings: A string is a sequence, len, Traversal with a for loop, String slices, Strings are immutable, Searching, Looping and Counting, String methods, The in operator, String comparison.

Case Study: Reading word lists, Search, Looping with indices.

Lists: List is a sequence, Lists are mutable, Traversing a list, List operations, List slices, List methods, Map filter and reduce, Deleting elements, Lists and Strings, Objects and values, Aliasing, List arguments.

Unit Outcomes:
Student should be able to

1. Use the data structure list.
2. Design programs for manipulating strings.

Unit – IV

Dictionaries: A dictionary is a mapping, Dictionary as a collection of counters, Looping and dictionaries, Reverse Lookup, Dictionaries and lists, Memos, Global Variables.

Tuples: Tuples are immutable, Tuple Assignment, Tuple as Return values, Variable-length argument tuples, Lists and tuples, Dictionaries and tuples, Sequences of sequences.

Files: Persistence, Reading and writing, Format operator, Filename and paths, Catching exceptions, Databases, Pickling, Pipes, Writing modules.

Classes and Objects: Programmer-defined types, Attributes, Instances as Return values, Objects are mutable, Copying.

Classes and Functions:

Unit Outcomes:
Student should be able to

1. Apply object orientation concepts.
2. Use data structure dictionaries.
3. Organize data in the form of files.

Unit – V

Classes and Functions: Time, Pure functions, Modifiers, Prototyping versus Planning

Classes and Methods: Object oriented features, Printing objects, The init method, The __str__method, Operator overloading, Type-based Dispatch, Polymorphism, Interface and Implementation

Inheritance: Card objects, Class attributes, Comparing cards, decks, Printing the Deck, Add Remove shuffle and sort, Inheritance, Class diagrams, Data encapsulation.

The Goodies: Conditional expressions, List comprehensions, Generator expressions, any and all, Sets, Counters, defaultdict, Named tuples, Gathering keyword Args,

Unit Outcomes:

Student should be able to

1. Plan programs using object orientation approach.
2. Illustrate the principle of inheritance.

Unit Outcomes:

Student should be able to

1. Apply the features of Python language in various real applications.
2. Select appropriate data structure of Python for solving a problem.
3. Design object oriented programs using Python for solving real-world problems.
4. Apply modularity to programs.

Text books:

1. Allen B. Downey, “Think Python”, 2nd edition, SPD/O’Reilly, 2016.

Reference Books:

1. Martin C.Brown, “The Complete Reference: Python”, McGraw-Hill, 2018.
2. Kenneth A. Lambert, B.L. Juneja, “Fundamentals of Python”, CENGAGE, 2015.
3. R. Nageswara Rao, “Core Python Programming”, 2nd edition, Dreamtech Press, 2019

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
3	0	0	3

19A05403T OPERATING SYSTEMS

(Common to CSE& IT)

Course Objectives:

The course is designed to

- Understand basic concepts and functions of operating systems
- Understand the processes, threads and scheduling algorithms.
- Provide good insight on various memory management techniques
- Expose the students with different techniques of handling deadlocks
- Explore the concept of file-system and its implementation issues
- Familiarize with the basics of Linux operating system
- Implement various schemes for achieving system protection and security

UNIT I

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Open-Source Operating Systems

System Structures:Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Operating system debugging, System Boot.

Unit Outcomes:

- Identify major components of operating systems
- Understand the types of computing environments
- Explore several open source operating systems
- Recognize operating system services to users, processes and other systems

UNIT II

Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems.
Multithreaded Programming: Multithreading models, Thread libraries, Threading issues, Examples.

Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling, Examples.
Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems - Dining philosophers problem, Readers and writers problem.

Unit Outcomes:

- Understand the importance, features of a process and methods of communication between processes.
- Improving CPU utilization through multi programming and multithreaded programming
- Examine several classical synchronization problems

UNIT III

Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation, Examples.
Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation, Examples.

Unit Outcomes:

- Examine the various techniques of allocating memory to processes
- Summarize how paging works in contemporary computer systems
- Understanding the benefits of virtual memory systems.

UNIT IV

Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection And recovery, Deadlock avoidance, Deadlock prevention.
File Systems: Files, Directories, File system implementation, management and optimization.
Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.

Unit Outcomes:

- Investigate methods for preventing/avoiding deadlocks
- Examine file systems and its interface in various operating systems
- Analyze different disk scheduling algorithms

UNIT V

System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights.

System Security: Introduction, Program threats, System and network threats, Cryptography as a security, User authentication, implementing security defenses, firewalling to protect systems and networks, Computer security classification.
Case Studies: Linux, Microsoft Windows.

Unit Outcomes:

- Infer various schemes available for achieving system protection.
- Acquiring knowledge about various countermeasures to security attacks
- Outline protection and security in Linux and Microsoft Windows.

Course Outcomes

By the end of this course students will be able to:

- Realize how applications interact with the operating system
- Analyze the functioning of a kernel in an Operating system.
- Summarize resource management in operating systems
- Analyze various scheduling algorithms
- Examine concurrency mechanism in Operating Systems
- Apply memory management techniques in design of operating systems
- Understand the functionality of file system
- Compare and contrast memory management techniques.
- Understand the deadlock prevention and avoidance.
- Perform administrative tasks on Linux based systems.

Text Books:

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2016.
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008. (Topics: Inter-process Communication and File systems.)

Reference Books:

1. Tanenbaum A S, Woodhull A S, Operating Systems Design and Implementation, 3rd edition, PHI, 2006.
2. Dhamdhere D M, Operating Systems A Concept Based Approach, 3rd edition, Tata McGraw-Hill, 2012.
3. Stallings W, Operating Systems -Internals and Design Principles, 6th edition, Pearson Education, 2009
4. Nutt G, Operating Systems, 3rd edition, Pearson Education, 2004

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
0	0	3	1.5

19A05304P PYTHON PROGRAMMING LAB
(Common to CSE & IT)

Course Objectives:

1. To train the students in solving computational problems
2. To elucidate solving mathematical problems using Python programming language
3. To understand the fundamentals of Python programming concepts and its applications.
4. To understand the object-oriented concepts using Python in problem solving.

Laboratory Experiments

1. Install Python Interpreter and use it to perform different Mathematical Computations. Try to do all the operations present in a Scientific Calculator
2. Write a function that draws a grid like the following:

```
+ - - - - + - - - +
|       |       |
|       |       |
|       |       |
|       |       |
+ - - - - + - - - +
|       |       |
|       |       |
|       |       |
|       |       |
+ - - - - + - - - - +
```

3. Write a function that draws a Pyramid with # symbols

```
          #
        # # #
      # # # # #
    # # # # # # #
          .
          .
          .
```

Up to 15 hashes at the bottom

4. Using turtles concept draw a wheel of your choice
5. Write a program that draws Archimedean Spiral
6. The letters of the alphabet can be constructed from a moderate number of basic elements, like vertical and horizontal lines and a few curves. Design an alphabet that can be drawn with a minimal number of basic elements and then write functions that draw the letters. The alphabet can belong to any Natural language excluding English. You should consider at least Ten letters of the alphabet.
7. The time module provides a function, also named time that returns the current Greenwich Mean Time in "the epoch", which is an arbitrary time used as a reference point. On UNIX systems, the epoch is 1 January 1970.

   >>> import time

   >>> time.time()

   1437746094.5735958

   Write a script that reads the current time and converts it to a time of day in hours, minutes, and seconds, plus the number of days since the epoch.
8. Given $n+r+1 <= 2^r$. n is the input and r is to be determined. Write a program which computes minimum value of r that satisfies the above.
9. Write a program that evaluates Ackermann function
10. The mathematician Srinivasa Ramanujan found an infinite series that can be used to generate a numerical approximation of $1/\pi$:

    Write a function called estimate_pi that uses this formula to compute and return an estimate of π.

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

It should use a while loop to compute terms of the summation until the last term is smaller than 1e-15 (which is Python notation for 10^{-15}). You can check the result by comparing it to math.pi.

11. Choose any five built-in string functions of C language. Implement them on your own in Python. You should not use string related Python built-in functions.
12. Given a text of characters, Write a program which counts number of vowels, consonants and special characters.
13. Given a word which is a string of characters. Given an integer say 'n', Rotate each character by 'n' positions and print it. Note that 'n' can be positive or negative.
14. Given rows of text, write it in the form of columns.
15. Given a page of text. Count the number of occurrences of each latter (Assume case insensitivity and don't consider special characters). Draw a histogram to represent the same
16. Write program which performs the following operations on list's. Don't use built-in functions
    a) Updating elements of a list
    b) Concatenation of list's
    c) Check for member in the list
    d) Insert into the list
    e) Sum the elements of the list
    f) Push and pop element of list
    g) Sorting of list
    h) Finding biggest and smallest elements in the list
    i) Finding common elements in the list

18. Write a program that reads a file, breaks each line into words, strips whitespace and punctuation from the words, and converts them to lowercase.

19. Go to Project Gutenberg (http://gutenberg.org) and download your favorite out-of-copyright book in plain text format. Read the book you downloaded, skip over the header information at the beginning of the file, and process the rest of the words as before. Then modify the program to count the total number of words in the book, and the number of times each word is used. Print

the number of different words used in the book. Compare different books by different authors, written in different eras.

20. Go to Project Gutenberg (http://gutenberg.org) and download your favorite out-of-copyright book in plain text format. Write a program that allows you to replace words, insert words and delete words from the file.

21. Consider all the files on your PC. Write a program which checks for duplicate files in your PC and displays their location. Hint: If two files have the same checksum, they probably have the same contents.

22. Consider turtle object. Write functions to draw triangle, rectangle, polygon, circle and sphere. Use object oriented approach.

23. Write a program illustrating the object oriented features supported by Python.

24. Design a Python script using the Turtle graphics library to construct a turtle bar chart representing the grades obtained by N students read from a file categorising them into distinction, first class, second class, third class and failed.

25. Design a Python script to determine the difference in date for given two dates in YYYY:MM:DD format(0 <= YYYY <= 9999, 1 <= MM <= 12, 1 <= DD <= 31) following the leap year rules.

26. Design a Python Script to determine the time difference between two given times in HH:MM:SS format.(0 <= HH <= 23, 0 <= MM <= 59, 0 <= SS <= 59)

Course outcomes:

Student should be able to

- Design solutions to mathematical problems.
- Organize the data for solving the problem.
- Develop Python programs for numerical and text based problems.
- Select appropriate programming construct for solving the problem.
- Illustrate object oriented concepts.

Reference Books:

1. Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, "How to Think Like a Computer Scientist: Learning with Python 3", 3rd edition, Available at http://www.ict.ru.ac.za/Resources/cspw/thinkcspy3/thinkcspy3.pdf
2. Paul Barry, "Head First Python a Brain Friendly Guide" 2nd Edition, O'Reilly, 2016
3. Dainel Y.Chen "Pandas for Everyone Python Data Analysis" Pearson Education, 2019

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-II Sem

L	T	P	C
0	0	3	1.5

19A05403P OPERATING SYSTEMS LAB

Course Objectives:

- To familiarize students with the architecture of OS.
- To provide necessary skills for developing and debugging CPU Scheduling algorithms.
- To elucidate the process management and scheduling and memory management.
- To explain the working of an OS as a resource manager, file system manager, process manager, memory manager, and page replacement tool.
- To provide insights into system calls, file systems and deadlock handling.

List of Experiments

1. Practicing of Basic UNIX Commands.
2. Write programs using following UNIX operating system calls
   Fork, exec, getpid, exit, wait, close, stst, opendir and readdir
3. Simulate UNIX commands like cp, ls, grep, etc.,
4. Simulate the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority
5. Implement dynamic priority scheduling algorithm.
6. Assume that there are five jobs with different weights ranging from 1 to 5. Implement round robin algorithm with time slice equivalent to weight.
7. Implement priority scheduling algorithm. While executing, no process should wait for more than 10 seconds. If waiting time is more than 10 seconds, that process has to be executed for atleast 1 second before waiting again.
8. Control the number of ports opened by the operating system with
   a) Semaphore b) Monitors.
9. Simulate how parent and child processes use shared memory and address space.
10. Simulate sleeping barber problem.
11. Simulate dining philosopher's problem.
12. Simulate producer and consumer problem using threads.
13. Implement the following memory allocation methods for fixed partition
    a) First fit b) Worst fit c) Best fit
14. Simulate the following page replacement algorithms
    a) FIFO b) LRU c) LFU etc.,
15. Simulate Paging Technique of memory management
16. Simulate Bankers Algorithm for Dead Lock avoidance and prevention

17. Simulate following file allocation strategies
    a) Sequential b) Indexed c) Linked
18. Simulate all File Organization Techniques
    a) Single level directory b) Two level c) Hierarchical d) DAG

Unit Outcomes:

- Trace different CPU Scheduling algorithm (L2).
- Implement Bankers Algorithms to Avoid and prevent the Dead Lock (L3).
- Evaluate Page replacement algorithms (L5).
- Illustrate the file organization techniques (L4).
- Illustrate shared memory process (L4).
- Design new scheduling algorithms (L6)

Reference Books:

1. Peter B. Galvin, Greg Gagne "Operating System Concepts", Abraham Silberchatz, Eighth Edition, John Wiley.
2. Stallings "Operating Systems: Internals and Design Principles", Sixth Edition–2009, Pearson Education
3. Andrew S Tanenbaum, "Modern Operating Systems", Second Edition, PHI.
4. S.Haldar, A.A.Aravind, "Operating Systems", Pearson Education.
5. B.L.Stuart, "Principles of Operating Systems", Cengage learning, India Edition.2013-2014
6. A.S.Godbole, "Operating Systems", Second Edition, TMH.
7. P.C.P. Bhatt, "An Introduction to Operating Systems", PHI.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B.Tech – II-I Sem

L	T	P	C
3	0	0	0

19A99301 ENVIRONMENTAL SCIENCE

Course Objectives:

- To make the students to get awareness on environment
- To understand the importance of protecting natural resources, ecosystems for future generations and pollution causes due to the day to day activities of human life
- To save earth from the inventions by the engineers.

UNIT – I

Multidisciplinary Nature Of Environmental Studies: – Definition, Scope and Importance – Need for Public Awareness.

Natural Resources : Renewable and non-renewable resources – Natural resources and associated problems – Forest resources – Use and over – exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems – Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies – Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies. – Energy resources:

Unit Outcomes

- To know the importance of public awareness
- To know about the various resources

UNIT – II

Ecosystems: Concept of an ecosystem. – Structure and function of an ecosystem – Producers, consumers and decomposers – Energy flow in the ecosystem – Ecological succession – Food chains, food webs and ecological pyramids – Introduction, types, characteristic features, structure and function of the following ecosystem:

a. Forest ecosystem.
b. Grassland ecosystem
c. Desert ecosystem
d. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

Biodiversity And Its Conservation : Introduction 0 Definition: genetic, species and ecosystem diversity – Bio-geographical classification of India – Value of biodiversity: consumptive use, Productive use, social, ethical, aesthetic and option values – Biodiversity at global, National and

local levels – India as a mega-diversity nation – Hot-sports of biodiversity – Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts – Endangered and endemic species of India – Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

Course Outcomes:

- To know about various echo systems and their characteristics
- To know about the biodiversity and its conservation

UNIT – III

Environmental Pollution: Definition, Cause, effects and control measures of :

a. Air Pollution.
b. Water pollution
c. Soil pollution
d. Marine pollution
e. Noise pollution
f. Thermal pollution
g. Nuclear hazards

Solid Waste Management : Causes**,** effects and control measures of urban and industrial wastes – Role of an individual in prevention of pollution – Pollution case studies – Disaster management: floods, earthquake, cyclone and landslides.

Course Outcomes:

- To know about the various sources of pollution.
- To know about the various sources of solid waste and preventive measures.
- To know about the different types of disasters and their managerial measures**.**

UNIT – IV

Social Issues And The Environment: From Unsustainable to Sustainable development – Urban problems related to energy – Water conservation, rain water harvesting, watershed management – Resettlement and rehabilitation of people; its problems and concerns. Case studies – Environmental ethics: Issues and possible solutions – Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case Studies – Wasteland reclamation. – Consumerism and waste products. – Environment Protection Act. – Air (Prevention and Control of Pollution) Act. – Water (Prevention and control of Pollution) Act – Wildlife Protection Act – Forest Conservation Act – Issues involved in enforcement of environmental legislation – Public awareness.

Course Outcomes:

- To know about the social issues related to environment and their protection acts.
- To know about the various sources of conservation of natural resources.
- To know about the wild life protection and forest conservation acts.

UNIT – V

Human Population And The Environment: Population growth, variation among nations. Population explosion – Family Welfare Programmes. – Environment and human health – Human Rights – Value Education – HIV/AIDS – Women and Child Welfare – Role of information Technology in Environment and human health – Case studies.

Field Work: Visit to a local area to document environmental assets River/forest grassland/hill/mountain – Visit to a local polluted site-Urban/Rural/Industrial/Agricultural Study of common plants, insects, and birds – river, hill slopes, etc..

Unit Outcomes:

- To know about the population explosion and family welfare programmes.
- To identify the natural assets and related case studies.

Course Outcomes:

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

- Grasp multidisciplinary nature of environmental studies and various renewable and nonrenewable resources.
- Understand flow and bio-geo- chemical cycles and ecological pyramids.
- Understand various causes of pollution and solid waste management and related preventive measures.
- About the rainwater harvesting, watershed management, ozone layer depletion and waste land reclamation.
- Casus of population explosion, value education and welfare programmes.

TEXT BOOKS :

1. Text book of Environmental Studies for Undergraduate Courses Erach Bharucha for University Grants Commission, Universities Press.
2. Palaniswamy, “Environmental Studies”, Pearson education
3. S.Azeem Unnisa, “Environmental Studies” Academic Publishing Company
4. K.Raghavan Nambiar, “Text book of Environmental Studies for Undergraduate Courses as per UGC model syllabus”, Scitech Publications(India), Pvt. Ltd.

REFERENCES :

1. Deeksha Dave and E.Sai Baba Reddy, “Textbook of Environmental Science”, Cengage Publications.
2. M.Anji Reddy, “Text book of Environmental Sciences and Technology”, BS Publication.
3. J.P.Sharma, Comprehensive Environmental studies, Laxmi publications.
4. J. Glynn Henry and Gary W. Heinke, “Environmental Sciences and Engineering”, Prentice hall of India Private limited
5. G.R.Chatwal, “A Text Book of Environmental Studies” Himalaya Pubilishing House
6. Gilbert M. Masters and Wendell P. Ela, “Introduction to Environmental Engineering and Science, Prentice hall of India Private limited.