PROGRAMME: B.E. Information Technology Engineering III-Semester

Analysis and Design of Algorithms

Course Objectives

Data structure includes analyzing various algorithms along with time and space complexities. It also helps students to design new algorithms through mathematical analysis and programming.

Unit-1: Introduction of Algorithms, Analysis of algorithms: Space Complexity, Time Complexity, recurrence relation and Asymptotic Notation, Divide and Conquer: General Methods, Analysis and Design, Binary Search, Quick sort, Merge sort, Strassen's matrix multiplication.

Unit-2: Greedy Strategy: Introduction, examples of greedy method like optimal merge pattern, Huffman coding, Minimum spanning trees, knapsack problem, job sequencing with dead lines single source shortest path algorithms.

Unit-3: Dynamic Programming: Introduction, Problem based on this approach such as 0/1 Knapsack Multistage graph, reliability design, Floyd-warshall algorithms.

Unit-4: Backtracking Concept and its example like 8 Queen's problem, Hamiltonian cycle, Graph coloring problem, 15 Puzzle problem, Least Cost Search

Unit-5: Introduction to branch & bound method, examples of branch & bound methods like traveling sales man problem, meaning of lower bound theory and its use in solving algebraic problem. NP-completeness & NP hard problems. Basic Concept of non deterministic algorithms. NP hard and NP complete classes.

Course Outcomes

1. Students will be able to understand fundamentals of algorithms.
2. Understanding various design methods for graphs.
3. Learning different concepts of backtracking including puzzle problem and graph coloring.
4. Getting familiar with non-deterministic algorithms and techniques of branch and bound.

Reference Books:

1. Horowitz, Sahani,Rajasekaran " Fundamentals of Computer Algorithms", Universities Press.
2. Thomas H. Cormen, "Introduction to Algorithms", PHI.
3. Harsh Bhasin "Algorithms Design and Analysis" Oxford**.**
4. I.Chandra Mohan " Design and Analysis of Algorithms" PHI

List of Experiments:

1. Implement Binary Search using C++.
2. Implement Quick sort using C++.
3. Implement Strassen Matrix multiplication on the given matrix.
4. Implement Merge sort on the given list of elements.
5. Implement Job sequencing problem using C++.
6. Implement Floyd warshall algorithm using C++.
7. Implement 8 – queens problem using backtracking.
8. Implement graph coloring problem using C++.
9. Implement 0/1 knapsack using branch and bound.
10. Implement travelling salesman problem using C++.

PROGRAMME: B.E. Information Technology Engineering III-Semester

Digital Circuits and Systems

Course Objectives

In the modern age electronic computers, communication systems and Internet became an important part of our life. The operation of these systems is based on the principle of digital techniques. The objective of this course is to get familiar with the concept of digital techniques and these systems are referred to as digital systems.

Unit-1: Number systems and logic gates: Decimal, Binary, Octal, Hexadecimal number systems and radix conversion. Codes- BCD, excess 3, gray, ASCII. Boolean algebra- Theorems and properties, Boolean functions, canonical and standard forms, De Morgans theorem, digital logic gates, Karnaugh maps.

Unit-2: Combinational circuits: Introduction to combinational circuits, multilevel NAND, NOR implementation. Designing binary Adders and Subtractors. Decoder, Encoder, Multiplexer, Demultiplexer circuits.

Unit-3: Sequential circuits: Introduction to Sequential circuits, flip-flops, RS, D, T, JK, M/S JK-flip-flops, truth tables, excitation tables and characteristic equations, clocked and edge triggered flip-flops, Registers- Definition, serial, parallel, shift left/right registers, Johnson counter, asynchronous and synchronous counters.

Unit-4: Digital logic families: Bipolar and unipolar logic families, Digital IC specifications, RTL, DTL, All types of TTL circuits, ECL, IIL, PMOS, NMOS & CMOS Logic.

Unit-5: Clocks and timing circuits: Bistable, Monostable & Astable multivibrator, Schmitt trigger circuit, Introduction of Analog to Digital & Digital to Analog converters, Display devices, 7 and 16 segment LED display, LCD.

Course Outcomes

On the completion of this course, students will be able to understand the basic building blocks of digital systems. The basic building block includes encoders, decoders, multiplexers and demultiplexers. These are commonly used in digital systems such as computers, communication systems and other modern technologies.

Reference Books:

1. M. Morris Mono, “Digital logic design”, Pearson Education Pvt. Ltd.
2. A Anand Kumar, “Fundamentals of digital circuits“, PHI Learning Pvt Ltd.
3. A K Maini, “Digital Electronics Principles and Integrated Circuits, Wiley India Pvt Ltd.
4. R P Jain, “Modern Digital Electronics”, Tata McGraw-Hill publishing company Ltd.
5. D P Kothari and J S Dhillon, "Digital Circuits and Design", Pearson Education Pvt. Ltd.

List of Experiments:

1. Study and verify the operation of AND, OR, NOT, NOR and NAND logic gates.
2. Design all basic logic gates using NOR universal gate.
3. Design all basic logic gates using NAND universal gate.
4. Verification of Demorgan’s theorem.
5. Construction and verification of half adder and full adder circuits.
6. Construction and verification of half subtractor and full subtractor circuits.
7. Design of Binary to Grey & Grey to Binary code Converters .
8. Design of BCD to excess-3 code converter.
9. Design and verification of Multiplexer circuit
10. Design and verification of De-multiplexer circuit.

PROGRAMME: B.E. Information Technology Engineering III-Semester

Object Oriented Programming & Methodology

Course Objectives

The objective of this course is to understand the advantage of object oriented programming over procedure oriented programming. It helps to understand the key features of Object Oriented Programming and Methodology like objects, methods, instance, message passing, encapsulation, polymorphism, data hiding, abstract data and inheritance.

Unit-1: Introduction: Object oriented programming, Introduction, Application, characteristics, difference between object oriented and procedure programming, Comparison of C and C++, Cout, Cin, Data Type, Type Conversion, Control Statement, Loops, Arrays and string arrays fundamentals, Function, Returning values from functions, Reference arguments, Overloaded function, Inline function, Default arguments, Returning by reference.

Unit-2: Object and Classes: Implementation of class and object in C++, access modifiers, object as data type, constructor, destructor, Object as function arguments, default copy constructor, parameterized constructor, returning object from function, Structures and classes, Classes objects and memory, static class data, Arrays of object, Arrays as class Member Data, The standard C++ String class, Run time and Compile time polymorphism.

Unit-3: Operator overloading and Inheritance: Overloading unary operators, Overloading binary operators, data conversion, pitfalls of operators overloading, Concept of inheritance, Derived class and base class, access modifiers, types of inheritance, Derived class constructors, member function, public and private inheritance.

Unit-4: Pointer and Virtual Function: Addresses and pointers, the address-of operator & pointer and arrays, Pointer and Function pointer, Memory management: New and Delete, pointers to objects, debugging pointers, Virtual Function, friend function, Static function, friend class, Assignment and copy initialization, this pointer, dynamic type information.

Unit-5: Streams and Files: Streams classes, Stream Errors, Disk File I/O with streams, file pointers, error handling in file I/O with member function, overloading the extraction and insertion operators, memory as a stream object, command line arguments, printer output, Function templates, Class templates Exceptions, Containers, exception handling.

Course Outcomes

On the completion of this course students will be able to:

1. Recognize attributes and methods for given objects.
2. Define data types but also deals with operations applied for data structures.
3. Implement algorithms and complex problems.

Reference Books:

1. E. Balaguruswami, ”Object Oriented Programming in C++”, TMH.
2. Robert Lafore, “Object Oriented Programming in C++”, Pearson.
3. M.T. Somashekare, D.S. Guru, “ Object-Oriented Programming with C++”, PHI.
4. Herbert Shildt, “The Complete Reference C++”, Tata McGraw Hill publication.

List of Experiments:

1. Write a program to find out the largest number using function.
2. Write a program to find the area of circle, rectangle and triangle using function overloading.
3. Write a program to implement complex numbers using operator overloading and type conversion.
4. Write a program using class and object to print bio-data of the students.
5. Write a program which defines a class with constructor and destructor which will count number of object created and destroyed.
6. Write a program to implement single and multiple inheritances taking student as the sample base class.
7. Write a program to add two private data members using friend function.
8. Write a program using dynamic memory allocation to perform 2x2 matrix addition and subtraction.
9. Write a program to create a stack using virtual function.
10. Write a program that store five student records in a file.
11. Write a program to get IP address of the system.
12. Write a program to shutdown the system on windows operating system.

PROGRAMME: B.E. Information Technology Engineering III-Semester

Discrete Structure

Course objectives

The main objectives of this course are:

1. To introduce students with sets, relations, functions, graph, and probability.
2. Students can perform set operation and solve logical reasoning and verify the correctness of logical statement.
3. They can apply the properties of relations and find partially ordered set and lattices.

Unit-1: Set Theory, Relation, Function, Theorem Proving Techniques : Set Theory: Definition of sets, countable and uncountable sets, Venn Diagrams, proofs of some general identities on sets Relation: Definition, types of relation, composition of relations, Pictorial representation of relation, Equivalence relation, Partial ordering relation, Job Scheduling problem Function: Definition, type of functions, one to one, into and onto function, inverse function, composition of functions, recursively defined functions, pigeonhole principle. Theorem proving Techniques: Mathematical induction, Proof by contradiction.

Unit-2: Algebraic Structures: Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results, Rings and Fields: definition and standard results.

Unit-3: Propositional Logic: Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Normal Forms, Universal and existential quantifiers. Introduction to finite state machine Finite state machines as models of physical system equivalence machines, Finite state machines as language recognizers

Unit-4: Graph Theory: Introduction and basic terminology of graphs, Planer graphs, Multigraphs and weighted graphs, Isomorphic graphs, Paths, Cycles and connectivity, Shortest path in weighted graph, Introduction to Eulerian paths and circuits, Hamiltonian paths and circuits, Graph coloring, chromatic number, Isomorphism and Homomorphism of graphs.

Unit-5: Posets, Hasse Diagram and Lattices: Introduction, ordered set, Hasse diagram of partially, ordered set, isomorphic ordered set, well ordered set, properties of Lattices, bounded and complemented lattices. Combinatorics: Introduction, Permutation and combination, Binomial Theorem, Recurrence Relation and Generating Function: Introduction to Recurrence Relation and Recursive algorithms , Linear recurrence relations with constant coefficients, Homogeneous solutions, Particular solutions, Total solutions , Generating functions , Solution by method of generating functions.

Course Outcomes

On completion of the course;

1. Students will be able to understand the notion of mathematical thinking, and algorithmic thinking and be able to apply them in problem solving such as formal specification, verification, and basic concepts of set theory.
2. Students understand the basic principle of Boolean algebra, logic and set theory.
3. Be able to construct simple mathematical proof and possess the ability to verify them.

Reference Books:

1. C.L.Liu" Elements of Discrere Mathematics" TMH.
2. Lipschutz, "Discrete mathematics (Schaum)",TMH.
3. U.S Gupta " Discrete Mathematical Structures" Pearson.
4. S. Santha," Discrete Mathematics with Combinatorics and graph theory", Cengage Learning.
5. Dr.Sukhendu. Dey " Graph Theory With Applications" Shroff Publishers