Madhav Institute of Technology & Science Gwalior (M.P.)
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade
Centre for Artificial Intelligence

DISCRETE STRUCTURES (3240221/ 3270221/ 3280221)

COURSE OBJECTIVES

- To perceive the knowledge of basic algebra
- To describe function and its relation
- To familiarize propositional logic
- To know about the graph theory and its application in computer engineering
- To familiarize the discrete numeric function and generating function

Unit I

Finite and Infinite Sets, Mathematical Induction, Principles of Inclusion and Exclusion, Multisets, Functions and Relations, Binary Relations, Equivalence Relations and Partitions, Partial Ordering Relations and Lattices, Chains, Pigeonhole Principle.

Unit II

Propositional Logic, Syntax, Semantics of ATF (Atomic Formula), WFF (Well Formed Formulas),Validity and Satisfiability of WFF by Quine's Method, Normal and Closure Form of Propositional Calculus.

Unit III

Introduction and Basic Terminology of Graphs, Planar Graphs, Multigraphs and Weighted Graph, Shortest Path in Weighted Graph, Introduction to Eulerian Paths and Circuits, Hamiltonian Paths and Circuits. Introduction to Trees, Rooted Trees, Path Length in Rooted Trees, Spanning Trees and Cut Trees.

Unit IV

Introduction to Discrete Numeric Functions and Generating Functions, Introduction to Recurrence Relations and Recursive Algorithms, Linear Recurrence Relations With Constant Coefficients, Homogeneous Solutions, Particular Solutions and Total Solutions.

Unit V

Introduction to Group, Subgroups, Generations and Evaluation of Power, Cosets and Lagrange's Theorem, Group Codes, Isomorphism and Automorphism, Homomorphism and Normal Sub Groups, Ring, Integral Domain and Field.

RECOMMENDED BOOKS

- J. Tremblay and Manohar: Discrete Mathematical Structures with Application to Computer science. Narsingh Deo: Graph Theory.
- Kenneth Rosen: Discrete mathematics and its applications (6th edition).2006. McGraw-Hill
- C. Liu, D. Mohapatra: Elements of Discrete Mathematics. 2008. Tata McGraw-Hill.
- T. Koshy: Discrete mathematics with applications.2003. Academic Press.
- J. Hein: Discrete structures, logic and computability.2009. Jones & Bartlett Publishers.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1: explain the concepts of set theory, propositional logic, graph theory, discrete numeric function and algebraic structure.

CO2: apply mathematical reasoning and logical thinking to solve problems

CO3: determine the solutions of problems pertaining to computer sciences using graph theory concepts.

CO4: solve counting problems using combinatorial analysis.

CO5: solve real-world problems by translating practical problems into mathematical formulations

CO6: formulate and design original mathematical proofs.

Madhav Institute of Technology & Science Gwalior (M.P.)
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade
Centre for Artificial Intelligence

MODERN COMPUTER ARCHITECTURE
(2240422/ 2270422/ 2280422/ 3240222/ 3270222/ 3280222)

COURSE OBJECTIVES

- To understand basic principles of Computer Systems.
- To understand various logic design techniques and their applications. They should be capable of using high performance computing architecture.

Unit I

Basic of Computer Architectures: Von Neumann Model, Various Subsystems, CPU, Memory, I/O, System Bus, Arithmetic Micro-Operation, Logic Micro Operation Shift Micro-Operation Register Transfer Micro Operations, Arithmetic, Micro-Operations, Logic Micro-Operations and Shift Micro-Operations.

Unit II

Multi-core Architecture: Memory technologies, hierarchical memory systems, the locality principle and caching, direct- mapped caches, block size, cache conflicts, associative caches, write strategies, advanced optimizations, performance improvement techniques, DRAM – organization, access techniques, scheduling algorithms, and signal systems.

Unit III

Distributed Computing Systems and Concurrency: Relation to Parallel Multiprocessors/multicomputer Systems, Distributed and Concurrent Programs, Message Passing vs. Shared Memory Systems, Synchronous vs. Asynchronous Executions, Design Issues and Challenges, Distributed Computing Technologies, Clocks and Synchronization, Global State and Distributed Transactions.

Unit IV

High Performance Computing (HPC): HPC Architecture, Parallel Processing, Parallel Memory Models, Data vs. Task Parallelism, High Throughput Computing, Vectorization, Multithreading.

Unit V

High Performance Computing with CUDA: CUDA programming model, Basic principles of CUDA programming, Concepts of threads and blocks, GPU and CPU data exchange

RECOMMENDED BOOKS

- M. Morris Mano, Computer System &Architecture, Prentice Hall of India, 2002.
- John L. Hennessy and David A Patterson, Computer Architecture-A quantitative approach, Morgan Kaufmann/ Elsevier, 4th Edition, 2007.
- Hayes. J.P, Computer architecture and organization by McGraw-Hill Companies, 1998
- Parallel Computer Architecture: A Hardware/Software Approach David Culler and J.P. Singh with Anoop Gupta, Morgan Kaufmann, 1998.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1: describe the organization of the Control unit, ALU, Memory and the I/O unit.

CO2: analyze different computer architectures and their applications.

CO3: contrast between different modes of Input-Output data transfer.

CO4: utilize the modern design structures of Pipelined and Multiprocessors systems.

CO5: evaluate the performance of distributed and high-performance computing architectures.

CO6: create parallel computing and programming models to harness the power of GPUs using CUDA.

Madhav Institute of Technology & Science Gwalior (M.P.)

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Centre for Artificial Intelligence

OBJECT ORIENTED PROGRAMMING (3240223/ 3270223/ 3280223)

COURSE OBJECTIVES

- To study the concept of object oriented programming.
- To create programs that leverage the object oriented features of the C++ Language.
- To apply object oriented programming techniques for real world problem solving.

Unit I

Introduction to Object Oriented Programming: Unstructured & Structured Programming, Object Oriented Paradigm, features and comparison with Procedural Oriented Programming approach. Specification of Class, Abstract Data Types, Visibility Modes, Defining Member Functions, Object Creation, Characteristics of Object, Scope Resolution Operator, Static Data Member, Static Member Function.

Unit II

Constructors and Destructors: Introduction, Types of Constructors- Default Constructor, User Defined Constructor, Parameterized Constructor, Copy Constructor, Constructor with Default Arguments, Rules of Constructor Definition and Usage, Destructors. Array of Objects, Object as Arguments, Inline Function, Friend Function.

Unit III

Polymorphism: Introduction, Type of Polymorphism: Compile Time Polymorphism & Run Time Polymorphism, Function Overloading, Operator Overloading: Binary and Unary Operators, Rules for Operator Overloading, Pitfalls of Operator Overloading, Data Conversion, Type Casting.

Unit IV

Inheritance: Introduction to Code Reuse, Visibility Modes, Types of Inheritance: Single Level, Multilevel, Multiple, Hybrid, Multipath. this Pointer, Ambiguity in MultiLevel Inheritance, Constructors in Derived Classes, Virtual Functions, Virtual Base Classes, Abstract Classes and Pure Virtual Functions, Nesting of Classes, Overriding Member Function, Association, Type of Association.

Unit V

Templates and Exceptions: Function Templates, Function Templates with Multiple Arguments, Class Templates, Exception Syntax, Exceptions with Arguments. Standard Template Library: Containers, Algorithms, Iterators, Vectors and Lists.

RECOMMENDED BOOKS

- Object Oriented Programming in C++, Robert Lafore, Sams.
- C++ How to Program, H M Deitel and P J Deitel, Prentice Hall.
- The Complete Reference in C++, Herbert Schildt, TMH.
- Fundamentals of Programming C++, Richard L. Halterman.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1: describe fundamental principles of object-oriented programming.

CO2: explain the benefits of object-oriented design.

CO3: develop well-structured, modular programs that leverage classes, objects, and inheritance to enhance code maintainability and reusability.

CO4: inspect the programs for the identification and correction of errors and exceptions.

CO5: apply standard template library for efficient and generic programming.

CO6: design effective solutions for real world problems using object-oriented principles.

DATA STRUCTURES (3240224/ 3270224/ 3280224)

COURSE OBJECTIVES

- To be familiar with the use of data structures as the foundational base for computer solutions to problems.
- To understand various techniques of searching and sorting.
- To understand basic concepts about stacks, queues, lists, trees and graphs.

Unit-I

Introduction to Data Structures: Algorithms & their characteristics, asymptotic notations. arrays and its representations, index to address translation. **Link list:** Introduction, implementation of linked list, operations, circular link list, doubly linked list, polynomial manipulation using linked list.

Unit-II

Stacks: Concepts and implementation of stacks, operations on stack, conversion of infix to postfix notation, evaluation of postfix expression, recursion.

Queues: Concepts and implementation, operations on queues, dequeue, priority queues, circular queues and application.

Unit-III

Trees: Types, terminology, binary tree -representations, traversal, conversion of general tree to binary tree, binary search tree, threaded binary tree and height balanced tree.

Unit-IV

Graphs: Background, graph theory terminologies, representation of graphs- sequential & linked representation, path matrix, graph traversals- BFS, DFS, spanning trees, applications of graph.

Unit-V

Searching & Sorting: Linear search, binary search, bubble sort, selection sort, insertion sort, quick sort, merge sort, radix sort and heap sort, comparison between sorting techniques, hashing and collision resolution techniques.

RECOMMENDED BOOKS

- Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Edition.
- An Introduction to Data Structures with Applications, Jean-Paul Tremblay, Mcgraw hill.
- Data Structures & Algorithms, Aho, Hopcroft & Ullman, original edition, Pearson Publication.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1: describe the basics of algorithms and their performance criteria.

CO2: explain the working of linear/non-linear data structures.

CO3: identify the appropriate data structure to solve specific problems.

CO4: decompose complex data structures, such as trees and graphs, into simpler components.

CO5: evaluate the time/space complexities of data structures & their applications.

CO6: design optimal algorithmic solutions for various real-world problems.

ELECTRONIC DEVICES AND CIRCUITS (3240225)

COURSE OBJECTIVES

- To impart the basic knowledge of the fundamentals of electronic circuits.
- To make familiarizing with semiconductor ICs and its applications
- To understand the system controlling schemes and their designing.

Unit I

PN Junction Diode: Ideal Diode, Depletion Layer, Diode Equation and I-V Characteristics, Static and Dynamic Resistance, Reverse Saturation Current, Zener and Avalanche Breakdown, Zener Diode, Zener Diode as Voltage Regulator, Rectifiers: Half Wave Rectifier, Full Wave Rectifiers (Center tapped and Bridge), Peak Inverse Voltage, Ripple Factor, Efficiency, Clipper and Clamper Circuits.

Unit II

Bipolar Junction Transistors: PNP and NPN Transistors, Common Base (CB), Common Emitter (CE) and Common Collector (CC) Configurations, Current Components in BJT, Current Gains: α, β and , Input and Output Characteristics in CB, CE and CC Modes, Transistor Biasing: Need for Biasing and Bias Stabilization, Load Line and Q-Point, Fixed Bias, Collector to Base Bias, Voltage Divider Bias and Emitter Bias. re-model and h-Parameter Equivalent Circuit of BJT, Small Signal Analysis of Single Stage CE Amplifier.

Unit III

Field Effect Transistors: Junction FET, Formation of Channel and Operating Principle, Pinch Off and Saturation Voltages and Currents, Drain and Transfer Characteristics of N-Channel JFET, FET Parameters. MOSFET (Construction, principal of Operation and symbol), MOSFET characteristics in Enhancement and Depletion modes.

Unit IV

Operational Amplifiers: Characteristics of Ideal and Practical Op-Amp, Open and Closed Loop Configuration, CMRR. Applications of Op-Amps: Inverting and Non-Inverting Amplifiers, Concept of Virtual Ground, Summing and Difference Amplifiers, Differentiator, Integrator, Multiplier and Divider, Logarithmic and Anti-logarithmic Amplifiers, Instrumentation amplifier. First order high pass & low pass Band pass filters, all pass filters and its Designing, PID controllers, Designing of PID controller circuit using OP-AMP.

Unit V

Introduction to Integrated Circuits, Classification, fundamental of monolithic IC technology, epitaxial

growth, masking and etching, diffusion of impurities. Realization of monolithic ICs and packaging. The 555 as monostable multivibrator, monostable multivibrator applications, The 555 as an astable multivibrator, a stable multivibrator application, Free running ramp generator. Applications of 555 Timer ICs.

RECOMMENDED BOOKS

- Robert Boylestad and Louis Nashelsky, " Electronic Device and Circuit Theory", PHI.
- Jacob Millman and Christos C. Halkias,and Satyabrata Jit "Millman's Electronic Device and Circuits", Tata McGraw- Hill.
- A.S.Sedra and K.C.Smith, "Microelectronic Circuits", Oxford University Press.
- David A.Bell, " Electronics Device and Circuits ", PHI.
- Ramakant A Gayakwad, "Op-Amps and Linear Integrated Circuits", Pearson.
- Franco, "Design with Operational Amplifiers and Analog ICs", McGraw Hill.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1: explain the structure and working principle of basic electronic devices.

CO2: design rectifiers and voltage regulators using diodes.

CO3: analyze the circuitry of BJTs and FETs.

CO4: develop biasing circuits using diodes and transistors.

CO5: design the algebraic, timer and supply circuits using Op-amps.

CO6: develop filter circuits through operational amplifiers.

OPERATING SYSTEMS (3270225/ 3280225)

COURSE OBJECTIVES

- To provide basic knowledge of computer operating system structures and functioning.
- To compare several different approaches to memory management, file management and process management.
- To understand various problems related to concurrent operations and their solutions.

Unit I

Basics of Operating System: Generations, Types, Structure, Services, System Calls, System Boot, System Programs, Protection and Security.

Unit II

Process Management: Process Concepts, Process States, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues.

Unit III

Process Synchronization: Background, Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors.

Deadlock: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock.

Unit IV

Memory Management: Main Memory, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

Unit V

Storage Management: Mass-Storage Structure, Overview, Disk Structure, Disk Attachment, Disk Scheduling.

File System Interface: The Concept of a File, Access Methods, Directory Structure, File System Structure, Allocation Methods, Free-Space Management.

RECOMMENDED BOOKS

- Operating System Concepts, Silberschatz, Ninth Edition, Willey Publication.
- Operating Systems, internals and Design Principles, Stallings, Seventh Edition, Pearson Publication.
- Modern Operating Systems, Tanenbaum, Fourth Edition. Pearson Publication.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1: illustrate the basic concepts of operating systems.

CO2: explain the working procedure of operating systems.

CO3: analyze various operating system problems and issues.

CO4: develop the solutions for various operating system problems and issues.

CO5: evaluate the performance of various scheduling and allocation techniques.

CO6: test the working of various scheduling and allocation techniques.

Madhav Institute of Technology & Science Gwalior (M.P.)
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade
Centre for Artificial Intelligence

ANNEXURE-IX

Experiment list/ Lab manual/ Skill Based Mini-Project List

of

Laboratory Courses offered

for

B. Tech II Semester

[Information Technology (Artificial Intelligence and Robotics)/ Artificial Intelligence (AI) and Data Science/ Artificial Intelligence (AI) and Machine Learning]

for batch admitted in 2023-24

(under the flexible curriculum)

Madhav Institute of Technology & Science Gwalior (M.P.)

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Centre for Artificial Intelligence

OBJECT ORIENTED PROGRAMMING (3240223/ 3270223/ 3280223)

List of Programs

1. Define a class representing a basic entity (e.g., a car, a person) with member variables and member functions. Create objects of the class and demonstrate their usage.
2. Implement a class with a constructor and destructor. Illustrate their roles and demonstrate how they are called during object creation and destruction.
3. Create a base class and a derived class. Demonstrate inheritance by inheriting properties and methods from the base class to the derived class using different types of inheritances.
4. Implement a polymorphic behavior using function overloading or overriding. Show how a function can behave differently based on the type of object.
5. Overload a few operators (e.g., +, -, *, =) for a custom class. Show how these operators can be used with objects of that class.
6. Implement a template class that can work with different data types. Demonstrate how to create objects of the template class with different data types.
7. Implement a program that uses exception handling to catch and handle runtime errors, such as division by zero or array out-of-bounds.
8. Utilize STL containers like vectors or lists in a program. Perform basic operations like insertion, deletion, and iteration.

Course Outcomes of Lab:

After completion of the course students will be able to:

CO1: apply the principles of object-oriented programming paradigm to programming problems.

CO2: use classes and objects to model real-world entities in a program.

CO3: apply the concepts of base classes, derived classes, and method overriding.

CO4: implement exception handling to manage errors and unexpected situations in a program.

CO5: create generic and reusable code using STL.

CO6: solve real world problems using the concepts of object-oriented programming.

Madhav Institute of Technology & Science Gwalior (M.P.)

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Centre for Artificial Intelligence

OBJECT ORIENTED PROGRAMMING (3240223/ 3270223/ 3280223)

List of Skill based Mini Projects

List of Micro Projects:

1. Develop a system to manage student information with classes for students, courses, and grades. Implement features like adding and removing students, enrolling in courses, and calculating GPA.
2. Create a system to manage inventory with classes for products, categories, and orders. Implement features like adding and removing products, updating stock levels, and processing orders.

List of Macro Projects:

1. Develop a system for managing hospital records with classes for patients, doctors, and appointments. Implement features like scheduling appointments, updating patient information, and generating reports.
2. Create a quiz game with classes for questions, quizzes, and players. Implement a scoring system, random question selection, and a leaderboard for tracking player performance.

List of Mini Projects:

1. Design a simple social media network with classes for users, posts, and comments. Implement features like creating posts, adding friends, and commenting on posts.
2. Develop a ticket booking system for a cinema or an event with classes for shows, venues, and tickets. Include features like booking tickets, checking seat availability, and generating tickets.

Madhav Institute of Technology & Science Gwalior (M.P.)

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Centre for Artificial Intelligence

DATA STRUCTURES (3240224/ 3270224/ 3280224)

List of Programs

1. Write a program to implement the following sorting algorithms on a list of integer:
    a. Selection Sort.
    b. Quick Sort
    c. Merge Sort
2. Write a program to implement stack and queue using arrays.
3. Write a program to count the number of nodes in a binary search tree.
4. Write a program to implement stack and queue using linked lists.
5. Write a program to implement AVL trees.
6. Write a program to implement a graph using arrays.
7. Write a program to implement a binary search algorithm.
8. Write a program to find the spanning tree of a graph.
9. Write a program to implement BFS and DFS graph traversal techniques.
10. Write a program to implement Heap sort.

Course Outcomes of Lab:

After completion of the course students will be able to:

CO1: implement sorting techniques.

CO2: analyze the complexity of tree data structure.

CO3: choose appropirate data structures for specific scenarios.

CO4: explain the importance of binary search tree.

CO5: solve real written problems using graphs.

Madhav Institute of Technology & Science Gwalior (M.P.)

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Centre for Artificial Intelligence

DATA STRUCTURES (3240224/ 3270224/ 3280224)

List of Skill based Mini Projects

Micro Projects:

- Use a heap to implement a priority queue.
- Develop a basic hash table with collision resolution

Macro Projects:

- Implement a program that can parse and evaluate mathematical expressions.
- Use a stack to implement an expression evaluator.

Mini Projects:

- Implement a Phone directory application using doubly linked lists.
- Implement a flight reservation system using a combination of data structures like priority queues for seat reservations and graphs for flight connections.